

Proyecto Final Tienda Online de Productos Electrónicos

Alumno: **Aaron Rodrigo Ramos Reyes**

Profesor: **Guillermo Monroy Rodríguez**

Materia: **Bases de Datos**

Universidad Autónoma Metropolitana

Unidad Cuajimalpa

1 de agosto de 2025

Índice

1. Introducción	2
1.1. Introducción al Proyecto	2
2. Diagrama ER y justificación de la normalización	2
2.1. Diagrama Entidad-Relación	2
2.2. Justificación de la normalización	3
3. Implementación de tablas en MySQL	4
3.1. Tabla Categoría	4
3.2. Tabla Producto	4
3.3. Tabla Cliente	5
3.4. Tabla Pedido	5
3.5. Tabla DetallePedido	6
3.6. Tabla Resena	6

1 Introducción

1.1 Introducción al Proyecto

El proyecto consiste en el diseño e implementación de una base de datos en **MySQL**, la cual debe cumplir con todos los requisitos descritos en el documento "*ProyectoFinalBD*".

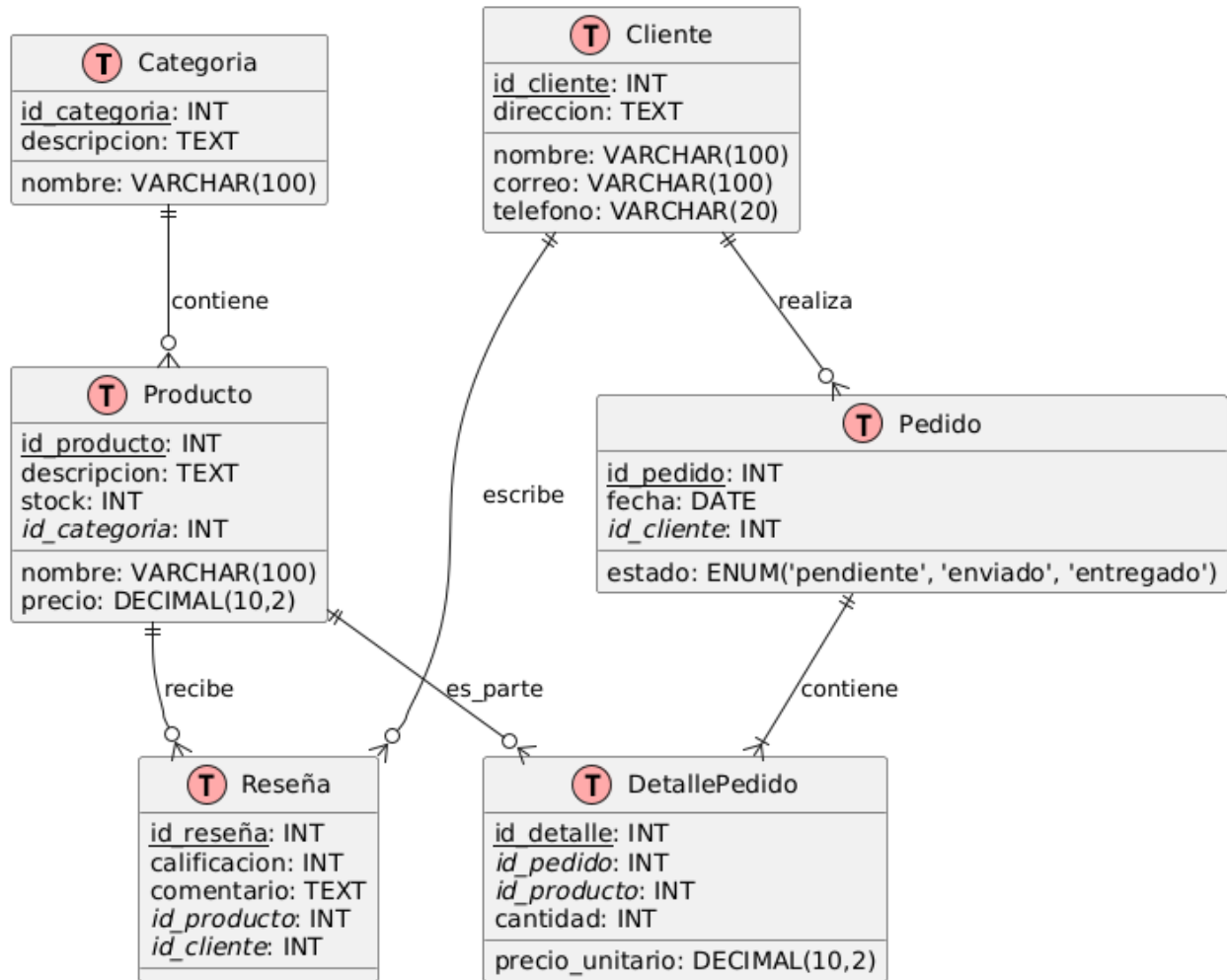
Con este propósito, se requerirá el uso y reforzamiento de tecnologías como **PlantUML** para la elaboración de diagramas, **L^AT_EX** para generar documentación clara, estructurada y modificable, y **MySQL** para la implementación de la base de datos. Asimismo, se utilizarán otras herramientas como los *stored procedures* para extender la funcionalidad de la base de datos.

Adicionalmente, se plantea la posibilidad de desarrollar un sistema en **Java** utilizando una arquitectura de tres capas, lo cual será opcional y dependerá del tiempo disponible para la entrega. Esta etapa adicional busca integrar y aplicar conocimientos adquiridos en otros cursos del plan de estudios.

2 Diagrama ER y justificación de la normalización

2.1 Diagrama Entidad-Relación

El siguiente diagrama muestra el modelo entidad-relación para el sistema de gestión de una tienda en línea de productos electrónicos. Se incluyen entidades como productos, clientes, pedidos, categorías, y reseñas, así como las relaciones entre ellas.



2.2 Justificación de la normalización

El modelo ha sido diseñado cumpliendo con las tres primeras formas normales para asegurar consistencia, evitar redundancia y facilitar el mantenimiento de los datos:

- **Primera Forma Normal (1FN):** Todas las columnas contienen valores atómicos y no hay grupos repetitivos.
- **Segunda Forma Normal (2FN):** No existen dependencias parciales; todos los atributos no clave dependen completamente de la clave primaria.
- **Tercera Forma Normal (3FN):** No hay dependencias transitivas; cada atributo no clave depende solamente de la clave primaria.

3 Implementación de tablas en MySQL

Este capítulo presenta la implementación del modelo entidad-relación en el sistema de gestión de bases de datos MySQL. Se creó una base de datos llamada **TiendaOnline** y a continuación se describen las tablas que la conforman.

3.1 Tabla Categoría

La tabla **Categoría** organiza los productos en grupos temáticos como teléfonos, laptops o accesorios. Contiene una clave primaria que la identifica de forma única.

```
CREATE TABLE Categoría (
  id_categoria INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  descripcion TEXT
);
```

Descripción de campos:

- **id_categoria:** Identificador único de la categoría.
- **nombre:** Nombre de la categoría (por ejemplo, "Laptops").
- **descripcion:** Descripción opcional de la categoría.

3.2 Tabla Producto

Almacena los productos disponibles en la tienda. Cada producto pertenece a una categoría.

```
CREATE TABLE Producto (
  id_producto INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  descripcion TEXT,
  precio DECIMAL(10,2) NOT NULL,
  stock INT NOT NULL CHECK (stock >= 0),
  id_categoria INT,
  FOREIGN KEY (id_categoria) REFERENCES Categoría(id_categoria)
);
```

Descripción de campos:

- **id_producto:** Identificador del producto.

- nombre, descripcion, precio: Informacion basica.
- stock: Inventario disponible, no puede ser negativo.
- id_categoria: Clave foranea que enlaza con la tabla Categoria.

3.3 Tabla Cliente

Contiene la informacion personal de cada cliente. El correo electronico debe ser unico.

```
CREATE TABLE Cliente (
  id_cliente INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  correo VARCHAR(100) NOT NULL UNIQUE,
  direccion TEXT,
  telefono VARCHAR(20)
);
```

Descripcion de campos:

- id_cliente: Identificador del cliente.
- correo: Sirve como clave candidata por ser unico.
- direccion, telefono: Informacion de contacto.

3.4 Tabla Pedido

Almacena las ordenes hechas por los clientes, incluyendo la fecha y el estado del pedido.

```
CREATE TABLE Pedido (
  id_pedido INT AUTO_INCREMENT PRIMARY KEY,
  fecha DATE NOT NULL,
  estado ENUM('pendiente', 'enviado', 'entregado') NOT NULL,
  id_cliente INT,
  FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente)
);
```

Descripcion de campos:

- id_pedido: Clave primaria del pedido.
- fecha: Fecha en que se realizo.
- estado: Estado actual del pedido (controlado por ENUM).
- id_cliente: Clave foranea hacia el cliente que realizo el pedido.

3.5 Tabla DetallePedido

Esta tabla representa una relacion muchos a muchos entre Pedido y Producto. Registra los productos incluidos en cada pedido, sus cantidades y precios al momento de la compra.

```
CREATE TABLE DetallePedido (  
  id_detalle INT AUTO_INCREMENT PRIMARY KEY,  
  id_pedido INT,  
  id_producto INT,  
  cantidad INT NOT NULL,  
  precio_unitario DECIMAL(10,2) NOT NULL,  
  FOREIGN KEY (id_pedido) REFERENCES Pedido(id_pedido),  
  FOREIGN KEY (id_producto) REFERENCES Producto(id_producto)  
);
```

Descripcion detallada:

- **id_detalle:** Identificador unico de cada linea del pedido.
- **id_pedido:** Clave foranea hacia el pedido correspondiente.
- **id_producto:** Clave foranea hacia el producto que se esta comprando.
- **cantidad:** Numero de unidades del producto en ese pedido.
- **precio_unitario:** Precio del producto en el momento del pedido (puede variar del precio actual).

Esta tabla es esencial para conservar un historico detallado de cada compra, permitiendo consultas como "¿que productos se compraron en el pedido 123?." "¿cuantas unidades de producto A se vendieron en total?".

3.6 Tabla Resena

Permite registrar las calificaciones y comentarios hechos por los clientes sobre los productos comprados.

```
CREATE TABLE Resena (  
  id_resena INT AUTO_INCREMENT PRIMARY KEY,  
  calificacion INT CHECK (calificacion BETWEEN 1 AND 5),  
  comentario TEXT,  
  id_producto INT,  
  id_cliente INT,  
  FOREIGN KEY (id_producto) REFERENCES Producto(id_producto),  
  FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente)  
);
```

Descripcion de campos:

- `id_resena`: Identificador de la resena.
- `calificacion`: Numero entero entre 1 y 5.
- `comentario`: Texto libre.
- `id_producto`, `id_cliente`: Claves foraneas para validar que el cliente resena un producto especifico.