



UNIVERSIDAD DE BUENOS AIRES - FACULTAD DE INGENIERÍA
2024 - 2do Cuatrimestre

Bases de Datos [75.15 / 95.05 / TA044]
Trabajo Práctico por equipos

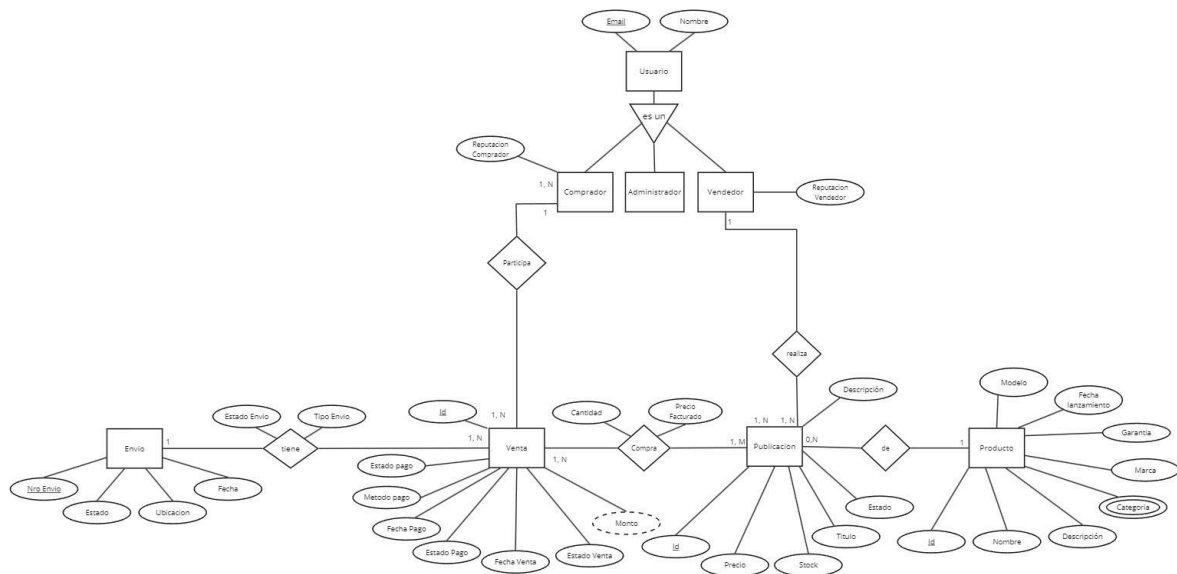
Docentes de la cátedra:
Cabrera Jose Luis y Bernardi Adrián Sergio

Integrantes del equipo:

Nombre Completo	Padrón	Correo
Mendoza, Manuel	107422	mamendoza@fi.uba.ar
Petean, Marina Zoe	110564	mpetean@fi.uba.ar

Fecha Límite de Entrega: 27 de Noviembre

Modelo Entidad Relación



Pasaje de Modelo Entidad Relación a Modelo Relacional

Tomamos inicialmente las entidades fuertes del MER y las convertimos en relaciones del modelo relacional, manteniendo sus claves primarias y atributos.

Elegimos utilizar herencia para modelar los roles y tener datos de la reputación de cada uno, debido a que cada Comprador y Vendedor posee una reputación independiente de la otra y un usuario puede tener ambos roles y, por lo tanto, tener dos reputaciones. Si bien el rol Administrador no posee reputación consideramos que podría mantenerse esa columna en NULL para esos casos porque serían menores en comparación a la cantidad de compradores y vendedores. Al realizar el pasaje a Modelo relacional, decidimos hacerlo con una relación “UsuarioRol” ya que no consideramos que los roles de Comprador, Administrador y Vendedor tengan atributos adicionales que los diferencien.

Un producto representa un objeto real específico que puede ser comprado y vendido por los usuarios de la plataforma, lo que implica que diferentes vendedores pueden ofrecer el mismo tipo de producto. Como el atributo categoría es multivaludado, y varios productos pueden compartir categorías, decidimos que en el modelo relacional se traduzca a tener una tabla “Categoría” con su id que la identifique y su nombre, y otra tabla que vincula a cada producto con la o las categorías a las que pertenece, llamada “ProductoCategoría” con la clave del producto y la clave de la categoría, así, un producto con N categorías, tendrá N apariciones en dicha tabla.

Una publicación es la relación que refleja la oferta del vendedor para permitir que un comprador adquiriera un producto determinado. En ella se registran el precio unitario actual del producto y la cantidad de stock disponible que posee el vendedor. Esta publicación es la que verá el comprador para seleccionar la cantidad de unidades que desea adquirir.

Para concretar una venta, creamos la tabla “Venta”, en la que se especifica el comprador, la publicación y otros datos relevantes. En el modelo Entidad-Relación, existía una relación de muchos a muchos entre la publicación y la venta (es decir, una venta puede incluir diferentes productos). Por ello, creamos una relación intermedia llamada “Detalle” que incluye ambas claves primarias, junto con la cantidad del producto comprado y el precio facturado. Además, dado que una venta tiene un único comprador, este se registra en la relación “Venta”.

La entidad “Envío” registra el proceso de envío de productos provenientes de diversas ventas, aunque cada venta se envía en un único envío. Sus atributos incluyen el estado y la ubicación actual del vehículo que transporta los productos. Al trasladarlo al modelo relacional, creamos una tabla intermedia llamada “DetalleEnvío”, que vincula la entidad “Envío” con la entidad “Venta”.

Esta relación permite diferenciar el envío general del estado de entrega de cada venta en particular, así como identificar si el comprador optó por el envío a domicilio o a un punto de retiro.

Entidades

1. Usuario

- Atributos:
 - id_usuario (PK)
 - nombre
 - email

2. Categoría

- Atributos:
 - id_categoria (PK)
 - nombre_categoria

3. CategoríaProducto

- Atributos:
 - id_categoria (PK)(FK -> Categoría.id_categoria)
 - id_producto (PK)(FK -> Producto.id_producto)

4. Producto

- Atributos:
 - id_producto (PK)
 - nombre
 - descripcion
 - modelo
 - marca
 - fecha_lanzamiento
 - garantia

5. Envío

- Atributos:

- nro_envio (PK)
- fecha
- estado_envio (Pendiente, Comenzado, Finalizado)
- ubicacion_actual

6. DetalleEnvio

- Atributos:

- id_envio (PK) (FK -> Envio.nro_envio)
- id_venta (PK) (FK -> Venta.id_venta)
- estado (Pendiente, Despachado, En camino, Entregado, Cancelado)
- tipo_envio (Domicilio, Punto entrega)

7. Venta

- Atributos:

- id_venta (PK)
- id_comprador (FK -> Usuario.id_usuario)
- estado (Concretada, Cancelada, En curso)
- fecha

8. Detalle

- Atributos:

- id_venta (PK) (FK -> Venta.id_venta)
- id_publicacion (PK) (FK -> Publicacion.id_publicacion)
- cantidad
- precioFacturado

9. PagoVenta

- Atributos:

- id_pago (PK)
- id_venta (FK -> Venta.id_venta)
- estado_pago (Pendiente, Pagado, Reembolsado)

- fecha
- metodo_pago (Efectivo, Transferencia, Credito, Debito)

10.UsuarioRol

- Atributos:
 - id_usuario (PK) (FK -> Usuario.id_usuario)
 - rol (PK) (Administrador, Vendedor, Comprador)
 - reputación (solo si es comprador o vendedor, sino NULL)

11.Publicación

- Atributos:
 - id_publicacion (PK)
 - id_vendedor (FK -> Usuario.id_usuario)
 - id_producto (FK -> Producto.id_producto)
 - precio
 - titulo
 - stock
 - estado
 - fecha_publicacion
 - descripción

Relaciones

Usuario (id_usuario, nombre, email)

UsuarioRol (id_usuario, rol, reputacion)

Producto (id_producto, nombre, descripcion, modelo, marca, color, fecha_lanzamiento, garantia)

Categoria (id_categoria, nombre_categoria)

CategoriaProducto (id_producto, id_categoria)

Publicacion (id_publicacion, id_vendedor, id_producto, titulo, descripcion, precio, stock, fecha_publicacion, estado)

Envio (nro_envio, fecha, estado_envio, ubicacion_actual)

Venta (id_venta, id_comprador, estado, fecha)

DetalleEnvio (nro_envio, id_venta, estado, tipo_envio)

Detalle (id_venta, id_publicacion, cantidad, precioFacturado)

PagoVenta (id_pago, id_venta, estado_pago, fecha, metodo_pago)

SQL:

Esquema.sql:

```
DROP DATABASE IF EXISTS TiendaOnline;
CREATE DATABASE TiendaOnline;
USE TiendaOnline;

CREATE TABLE Usuario (
    id_usuario BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL
);

CREATE TABLE UsuarioRol(
    id_usuario BIGINT UNSIGNED,
    rol ENUM('Administrador', 'Vendedor', 'Comprador') NOT NULL,
    reputacion INT DEFAULT 0 CHECK (reputacion >= 0),
    PRIMARY KEY (id_usuario, rol),
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)
);

CREATE TABLE Producto (
    id_producto BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT NOT NULL,
    modelo VARCHAR(100) NOT NULL,
    marca VARCHAR(50),
    color VARCHAR(20),
    fecha_lanzamiento DATE,
    garantia VARCHAR(100)
);

CREATE TABLE Categoria (
    id_categoria BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    nombre_categoria VARCHAR(50) NOT NULL
);

CREATE TABLE CategoriaProducto (
    id_producto BIGINT UNSIGNED,
    id_categoria BIGINT UNSIGNED,
    PRIMARY KEY (id_producto, id_categoria),
    FOREIGN KEY (id_producto) REFERENCES Producto(id_producto),
    FOREIGN KEY (id_categoria) REFERENCES Categoria(id_categoria)
);
```



```

CREATE TABLE Publicacion (
    id_publicacion BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    id_vendedor BIGINT UNSIGNED,
    id_producto BIGINT UNSIGNED,
    titulo VARCHAR(255) NOT NULL,
    descripcion TEXT,
    precio DECIMAL(10, 2) NOT NULL CHECK (precio > 83),
    stock INT DEFAULT 1,
    fecha_publicacion DATETIME DEFAULT CURRENT_TIMESTAMP,
    estado ENUM('Activa', 'Pausada', 'Finalizada') DEFAULT
'Activa',
    FOREIGN KEY (id_vendedor) REFERENCES Usuario(id_usuario),
    FOREIGN KEY (id_producto) REFERENCES Producto(id_producto)
);

```

```

CREATE TABLE Envio (
    nro_envio BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    fecha TIMESTAMP NOT NULL,
    estado_envio ENUM('Pendiente', 'Comenzado', 'Finalizado',
'Cancelado') DEFAULT 'Pendiente',
    ubicacion_actual VARCHAR(255)
);

```

```

CREATE TABLE Venta (
    id_venta BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    id_comprador BIGINT UNSIGNED,
    estado ENUM('Concretada', 'Cancelada', 'En curso') DEFAULT
'En curso',
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (id_comprador) REFERENCES Usuario(id_usuario)
);

```

```

CREATE TABLE DetalleEnvio (
    nro_envio BIGINT UNSIGNED,
    id_venta BIGINT UNSIGNED,
    estado ENUM('Pendiente', 'Despachado', 'En camino',
'Cancelado') DEFAULT 'Pendiente',
    tipo_envio ENUM('Domicilio', 'Punto entrega'),
    FOREIGN KEY (nro_envio) REFERENCES Envio(nro_envio),
    FOREIGN KEY (id_venta) REFERENCES Venta(id_venta)
);

```

```

CREATE TABLE Detalle (
    id_venta BIGINT UNSIGNED,
    id_publicacion BIGINT UNSIGNED,

```

```

    cantidad INT CHECK (cantidad > 0),
    precioFacturado DECIMAL(10, 2) CHECK (precioFacturado > 0),
    PRIMARY KEY (id_venta, id_publicacion),
    FOREIGN KEY (id_venta) REFERENCES Venta(id_venta),
    FOREIGN KEY (id_publicacion) REFERENCES
Publicacion(id_publicacion)
);

```

```

CREATE TABLE PagoVenta (
    id_pago BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    id_venta BIGINT UNSIGNED,
    estado_pago ENUM('Pendiente', 'Pagado', 'Reembolsado') DEFAULT
'Pendiente',
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    metodo_pago ENUM('Efectivo', 'Transferencia', 'Credito',
'Debito') NOT NULL
);

```

```

DELIMITER $$

```

```

CREATE TRIGGER trigger_actualizar_reputacion_vendedor
AFTER UPDATE ON Venta
FOR EACH ROW
BEGIN
    IF NEW.estado = 'Concretada' THEN
        UPDATE UsuarioRol
        SET reputacion = (
            SELECT COUNT(*)
            FROM Venta v
            INNER JOIN Detalle dp ON v.id_venta = dp.id_venta
            INNER JOIN Producto p ON dp.id_producto =
p.id_producto
            WHERE p.id_vendedor = NEW.id_comprador
            AND v.estado = 'Concretada'
        )
        WHERE id_usuario = NEW.id_comprador AND rol = 'Vendedor';
    END IF;
END$$

```

```

DELIMITER $$

```

```

CREATE TRIGGER trigger_actualizar_reputacion_comprador
AFTER UPDATE ON PagoVenta
FOR EACH ROW
BEGIN
    IF NEW.estado_pago = 'Pagado' THEN
        UPDATE UsuarioRol
        SET reputacion = (

```

```

        SELECT COUNT(*)
        FROM Venta v
        WHERE v.id_comprador = (
            SELECT id_comprador
            FROM Venta
            WHERE id_venta = NEW.id_venta
        )
        AND v.estado = 'Concretada'
    )
    WHERE id_usuario = (
        SELECT id_comprador
        FROM Venta
        WHERE id_venta = NEW.id_venta
    )
    AND rol = 'Comprador';
END IF;
END$$

DELIMITER ;

```

Permisos.sql:

```

DROP ROLE IF EXISTS Administrador;
DROP ROLE IF EXISTS Vendedor;
DROP ROLE IF EXISTS Comprador;

CREATE ROLE Administrador;
CREATE ROLE Vendedor;
CREATE ROLE Comprador;

GRANT ALL PRIVILEGES ON *.* TO Administrador WITH GRANT OPTION;

REVOKE ALL PRIVILEGES ON *.* FROM 'Vendedor';
REVOKE ALL PRIVILEGES ON *.* FROM 'Comprador';
FLUSH PRIVILEGES;

CREATE VIEW VistaPublicacion AS
SELECT titulo, precio, descripcion, stock
FROM TiendaOnline.Publicacion;

CREATE VIEW VistaEnvioComprador AS
SELECT tipo_envio, estado
FROM TiendaOnline.DetalleEnvio;

GRANT SELECT, INSERT, UPDATE ON TiendaOnline.CategoriaProducto TO
Vendedor;
GRANT SELECT, INSERT, UPDATE ON TiendaOnline.Categoria TO
Vendedor;

```

```
GRANT SELECT, INSERT, UPDATE ON TiendaOnline.VistaPublicacion TO Vendedor;
```

```
GRANT SELECT ON TiendaOnline.VistaPublicacion TO Comprador;  
GRANT INSERT, UPDATE ON TiendaOnline.Venta TO Comprador;  
GRANT INSERT, UPDATE ON TiendaOnline.Detalle TO Comprador;  
GRANT SELECT, UPDATE ON TiendaOnline.VistaEnvioComprador TO Comprador;  
GRANT INSERT, UPDATE ON TiendaOnline.Venta TO Comprador;  
GRANT SELECT ON TiendaOnline.Envio TO Comprador;
```

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'clave_admin';  
CREATE USER 'vendedor'@'localhost' IDENTIFIED BY 'clave_vendedor';  
CREATE USER 'comprador'@'localhost' IDENTIFIED BY 'clave_comprador';
```

```
GRANT Vendedor TO 'root'@'localhost';  
GRANT Comprador TO 'root'@'localhost';  
GRANT Administrador TO 'root'@'localhost';
```

```
GRANT Vendedor TO 'vendedor'@'localhost';  
GRANT Comprador TO 'comprador'@'localhost';  
GRANT Administrador TO 'admin'@'localhost';
```

```
SET DEFAULT ROLE Administrador TO 'admin'@'localhost';  
SET DEFAULT ROLE Vendedor TO 'vendedor'@'localhost';  
SET DEFAULT ROLE Comprador TO 'comprador'@'localhost';
```

```
SHOW GRANTS FOR 'admin'@'localhost';  
SHOW GRANTS FOR 'vendedor'@'localhost';  
SHOW GRANTS FOR 'comprador'@'localhost';
```

Inserts.sql:

```
INSERT INTO Usuario (nombre, email) VALUES  
('Juan Pérez', 'juan.perez@example.com'),  
('Ana Gómez', 'ana.gomez@example.com'),  
('Carlos Rodríguez', 'carlos.rodriguez@example.com'),  
('Laura Fernández', 'laura.fernandez@example.com'),  
('Pedro Martínez', 'pedro.martinez@example.com'),  
('Marta Sánchez', 'marta.sanchez@example.com'),  
('Luis García', 'luis.garcia@example.com'),  
('Patricia Torres', 'patricia.torres@example.com'),  
('José López', 'jose.lopez@example.com'),  
('Beatriz Díaz', 'beatriz.diaz@example.com'),  
('Daniel Ruiz', 'daniel.ruiz@example.com'),  
('Sofía Morales', 'sofia.morales@example.com'),  
('Andrés Castillo', 'andres.castillo@example.com'),
```

```
('Claudia Rivas', 'claudia.rivas@example.com'),  
( 'Ricardo Vega', 'ricardo.vega@example.com'),  
( 'Elena Ortiz', 'elena.ortiz@example.com'),  
( 'Gabriel Herrera', 'gabriel.herrera@example.com'),  
( 'Isabel Cruz', 'isabel.cruz@example.com'),  
( 'Manuel Navarro', 'manuel.navarro@example.com'),  
( 'María León', 'maria.leon@example.com');
```

```
INSERT INTO UsuarioRol (id_usuario, rol, reputacion) VALUES  
(1, 'Administrador', 10),  
(2, 'Vendedor', 5),  
(2, 'Comprador', 5),  
(3, 'Comprador', 3),  
(3, 'Vendedor', 3),  
(4, 'Vendedor', 6),  
(5, 'Comprador', 2),  
(6, 'Vendedor', 4),  
(7, 'Comprador', 7),  
(8, 'Administrador', 8),  
(9, 'Comprador', 1),  
(10, 'Vendedor', 9),  
(11, 'Vendedor', 5),  
(11, 'Comprador', 2),  
(12, 'Comprador', 2),  
(13, 'Administrador', 9),  
(14, 'Vendedor', 4),  
(15, 'Comprador', 3),  
(16, 'Administrador', 7),  
(17, 'Vendedor', 6),  
(17, 'Comprador', 7),  
(18, 'Comprador', 1),  
(19, 'Vendedor', 8),  
(20, 'Administrador', 10);
```

```
INSERT INTO Producto (nombre, descripcion, modelo, marca, color,  
fecha_lanzamiento, garantia) VALUES  
( 'Laptop HP', 'Laptop HP con 16GB de RAM y procesador i7',  
'HP-2024', 'HP', 'Negro', '2024-10-01', '2 años'),  
( 'Smartphone Samsung', 'Smartphone Samsung Galaxy S23', 'S23',  
'Samsung', 'Azul', '2024-01-15', '1 año'),  
( 'Auriculares Bose', 'Auriculares Bose con cancelación de ruido',  
'Bose-100', 'Bose', 'Blanco', '2024-03-10', '3 años'),  
( 'Televisor LG', 'Televisor 4K de 55 pulgadas', 'LG-55UHD', 'LG',  
'Negro', '2024-06-20', '5 años'),  
( 'Tablet Apple', 'Tablet Apple iPad Pro 12.9', 'iPadPro2024',  
'Apple', 'Gris', '2024-09-01', '2 años'),  
( 'Smartwatch Garmin', 'Reloj inteligente Garmin Fenix 7',  
'Fenix7', 'Garmin', 'Negro', '2024-02-25', '1 año'),
```

```

('Auriculares Sony', 'Auriculares Sony WH-1000XM5', 'Sony-1000',
'Sony', 'Negro', '2024-05-15', '2 años'),
('Smartphone Xiaomi', 'Smartphone Xiaomi Mi 12', 'Mi12', 'Xiaomi',
'Rojo', '2024-08-10', '1 año'),
('Laptop Dell', 'Laptop Dell XPS 13 con 8GB de RAM', 'XPS13',
'Dell', 'Plata', '2024-11-01', '3 años'),
('Monitor Acer', 'Monitor Acer Predator de 27 pulgadas',
'Predator-27', 'Acer', 'Negro', '2024-07-15', '2 años'),
('Cámara Canon', 'Cámara réflex Canon EOS R5', 'EOS-R5', 'Canon',
'Negro', '2024-03-01', '2 años'),
('Impresora Epson', 'Impresora Epson EcoTank L3250', 'L3250',
'Epson', 'Blanco', '2024-04-10', '1 año'),
('Router TP-Link', 'Router inalámbrico TP-Link Archer AX73',
'AX73', 'TP-Link', 'Negro', '2024-02-20', '3 años'),
('Disco Duro Seagate', 'Disco duro externo Seagate de 2TB',
'Expansion-2TB', 'Seagate', 'Negro', '2024-05-05', '2 años'),
('Teclado Logitech', 'Teclado mecánico inalámbrico Logitech MX
Keys', 'MXKeys', 'Logitech', 'Gris', '2024-06-18', '2 años'),
('Mouse Razer', 'Mouse gamer Razer DeathAdder V3',
'DeathAdder-V3', 'Razer', 'Negro', '2024-07-20', '2 años'),
('Parlante JBL', 'Parlante portátil JBL Charge 5', 'Charge5',
'JBL', 'Azul', '2024-01-25', '1 año'),
('Microondas Panasonic', 'Microondas Panasonic NN-SN936B de 2.2
pies cúbicos', 'NN-SN936B', 'Panasonic', 'Negro', '2024-08-05', '3
años'),
('Frigorífico Samsung', 'Frigorífico Samsung French Door
RF28R7201', 'RF28R7201', 'Samsung', 'Plata', '2024-09-15', '10
años'),
('Bicicleta Trek', 'Bicicleta de montaña Trek Marlin 7',
'Marlin7', 'Trek', 'Rojo', '2024-10-10', '5 años');

```

```

INSERT INTO Categoria (nombre_categoria) VALUES
('Electrónica'),
('Tecnología'),
('Audio'),
('Hogar'),
('Gadgets'),
('Deportes'),
('Electrodomésticos'),
('Moda'),
('Juguetes'),
('Muebles'),
('Vehículos'),
('Libros');

```

```

INSERT INTO CategoriaProducto (id_producto, id_categoria) VALUES
(1, 1), (1, 2),
(2, 1), (2, 2),

```

```
(4, 1), (4, 4),
(5, 2), (5, 1),
(6, 5), (10, 4),
(7, 3), (8, 1),
(9, 2), (10, 1),
(11, 7), (11, 1),
(12, 2), (12, 8),
(13, 3), (13, 9),
(14, 7), (14, 10),
(15, 2), (15, 12),
(16, 6), (16, 5),
(17, 3), (17, 1),
(18, 1), (18, 11),
(19, 2), (19, 1),
(20, 2), (20, 11),
(3, 3);
```

```
INSERT INTO Publicacion (id_vendedor, id_producto, titulo,
descripcion, precio, stock, estado) VALUES
(2, 1, 'Laptop HP 16GB', 'Laptop HP con procesador Intel i7 y 16GB
de RAM', 800.00, 10, 'Activa'),
(2, 2, 'Smartphone Samsung Galaxy S23', 'Smartphone Samsung S23,
pantalla 6.1" y 128GB de almacenamiento', 999.99, 5, 'Activa'),
(3, 3, 'Auriculares Bose', 'Auriculares Bose con cancelación
activa de ruido', 350.00, 15, 'Activa'),
(4, 4, 'Televisor LG 55" 4K', 'Televisor LG con resolución 4K y
HDR', 1200.00, 8, 'Activa'),
(5, 5, 'Tablet Apple iPad Pro 12.9', 'Tablet Apple con pantalla
Retina y 128GB de almacenamiento', 1100.00, 12, 'Activa'),
(6, 6, 'Smartwatch Garmin Fenix 7', 'Smartwatch Garmin Fenix 7 con
GPS y sensor de frecuencia cardíaca', 600.00, 20, 'Activa'),
(7, 7, 'Auriculares Sony WH-1000XM5', 'Auriculares Sony con
cancelación de ruido y conectividad Bluetooth', 400.00, 25,
'Activa'),
(8, 8, 'Smartphone Xiaomi Mi 12', 'Smartphone Xiaomi con 8GB de
RAM y pantalla AMOLED', 750.00, 18, 'Activa'),
(9, 9, 'Laptop Dell XPS 13', 'Laptop Dell con procesador Intel i5
y 8GB de RAM', 900.00, 14, 'Activa'),
(10, 10, 'Monitor Acer Predator 27"', 'Monitor Acer Predator de
27" con tasa de refresco de 144Hz', 500.00, 30, 'Activa'),
(1, 11, 'Lavadora Samsung 10kg', 'Lavadora de carga frontal con
capacidad de 10kg', 450.00, 5, 'Activa'),
(2, 12, 'Reloj Casio Vintage', 'Reloj digital estilo retro con
cronómetro', 30.00, 20, 'Activa'),
(3, 13, 'Pelota Adidas', 'Pelota de fútbol Adidas oficial de la
FIFA', 50.00, 15, 'Activa'),
(4, 14, 'Sofá 3 plazas', 'Sofá cómodo y moderno con capacidad para
3 personas', 600.00, 3, 'Activa'),
```

```
(5, 15, 'Libro: El Principito', 'Edición especial del clásico
libro "El Principito"', 20.00, 40, 'Activa'),
(6, 16, 'Auriculares Bluetooth JBL', 'Auriculares con batería de
larga duración', 120.00, 10, 'Activa'),
(7, 17, 'Barra de Sonido Sony', 'Barra de sonido con subwoofer
inalámbrico', 300.00, 8, 'Activa'),
(8, 18, 'Bicicleta Eléctrica Xiaomi', 'Bicicleta eléctrica
plegable Xiaomi con autonomía extendida', 1200.00, 4, 'Activa'),
(9, 19, 'Cámara GoPro Hero 10', 'Cámara GoPro con grabación en 5K
y resistente al agua', 500.00, 12, 'Activa'),
(10, 20, 'Patineta Eléctrica Razor', 'Patineta eléctrica con
velocidad máxima de 25 km/h', 400.00, 6, 'Activa');
```

```
INSERT INTO Envio (fecha, estado_envio, ubicacion_actual) VALUES
('2024-11-20 10:00:00', 'Pendiente', 'Almacén Central'),
('2024-11-21 11:30:00', 'Comenzado', 'Ruta de distribución'),
('2024-11-22 14:45:00', 'Finalizado', 'Dirección del comprador'),
('2024-11-23 09:00:00', 'Pendiente', 'Almacén 2'),
('2024-11-24 08:30:00', 'Pendiente', 'Almacén 1'),
('2024-11-25 12:00:00', 'Pendiente', 'Almacén Central'),
('2024-11-26 15:45:00', 'Comenzado', 'Ruta de distribución'),
('2024-11-27 17:20:00', 'Finalizado', 'Dirección del comprador'),
('2024-11-28 10:30:00', 'Pendiente', 'Almacén Secundario');
```

```
INSERT INTO Venta (id_comprador, estado) VALUES
(3, 'En curso'),
(4, 'Concretada'),
(5, 'En curso'),
(6, 'Concretada'),
(7, 'En curso'),
(8, 'Concretada'),
(9, 'En curso'),
(10, 'Concretada'),
(11, 'En curso'),
(12, 'Concretada'),
(13, 'En curso'),
(14, 'Concretada'),
(15, 'En curso');
```

```
INSERT INTO DetalleEnvio (nro_envio, id_venta, estado, tipo_envio)
VALUES
(1, 1, 'Pendiente', 'Domicilio'),
(2, 2, 'Despachado', 'Punto entrega'),
(3, 3, 'En camino', 'Domicilio'),
(4, 4, 'Pendiente', 'Punto entrega'),
(5, 5, 'Pendiente', 'Domicilio'),
(6, 9, 'Pendiente', 'Domicilio'),
(7, 10, 'En camino', 'Punto entrega'),
```



```
(8, 11, 'Despachado', 'Domicilio');
```

```
INSERT INTO Detalle (id_venta, id_publicacion, cantidad,  
precioFacturado) VALUES
```

```
(1, 1, 1, 800.00),  
(1, 2, 2, 999.99),  
(1, 3, 1, 350.00),  
(2, 4, 1, 1200.00),  
(2, 1, 1, 1100.00),  
(2, 6, 1, 600.00),  
(2, 2, 2, 400.00),  
(3, 5, 2, 2200.00),  
(3, 6, 1, 600.00),  
(4, 7, 1, 400.00),  
(5, 8, 2, 1500.00),  
(6, 9, 1, 900.00),  
(6, 10, 1, 500.00),  
(7, 1, 1, 800.00),  
(8, 2, 1, 999.99),  
(8, 3, 3, 1050.00);
```

```
INSERT INTO PagoVenta (id_venta, estado_pago, metodo_pago) VALUES
```

```
(1, 'Pendiente', 'Efectivo'),  
(2, 'Pagado', 'Transferencia'),  
(3, 'Pendiente', 'Credito'),  
(4, 'Pagado', 'Debito'),  
(5, 'Pendiente', 'Efectivo'),  
(6, 'Pagado', 'Transferencia'),  
(7, 'Pendiente', 'Credito'),  
(8, 'Pagado', 'Debito'),  
(9, 'Pendiente', 'Efectivo'),  
(10, 'Pagado', 'Transferencia'),  
(9, 'Pendiente', 'Credito'),  
(10, 'Pagado', 'Efectivo'),  
(11, 'Pendiente', 'Transferencia'),  
(12, 'Pagado', 'Debito'),  
(13, 'Pendiente', 'Efectivo');
```

Consultas.sql:

```
-- Listar todos los productos de la plataforma
```

```
SELECT * FROM TiendaOnline.Producto;
```

```
-- Listar productos de la plataforma por categoría
```

```
SELECT *
```

```
FROM TiendaOnline.Producto p, TiendaOnline.CategoriaProducto cp,
```

```
TiendaOnline.Categoria c
```

```
WHERE c.id_categoria = cp.id_categoria
```

```

AND p.id_producto = cp.id_producto
AND c.nombre_categoria = 'Tecnología';

-- Crear una publicación para vender
INSERT INTO TiendaOnline.Publicacion (id_vendedor, id_producto,
titulo, descripcion, precio, stock)
VALUES (2, 1, 'Laptop HP NUEVA', 'Laptop muy linda', 500000.00,
2);

-- Actualizar una publicación de venta
UPDATE TiendaOnline.Publicacion
SET precio = 4500.00
WHERE id_publicacion = 11;

-- Realizar una compra (Creo la venta(El monto se calcula con el
detalle))
INSERT INTO TiendaOnline.Venta (id_comprador)
VALUES (2);
-- (Agrego productos a la venta)
INSERT INTO TiendaOnline.Detalle (id_venta, id_publicacion,
cantidad, precioFacturado)
VALUES ((SELECT MAX(id_venta) FROM Venta), 11, 1, 4500.00);

-- Guardar un envío
INSERT INTO TiendaOnline.Envio (fecha, ubicacion_actual)
VALUES ('2024-12-12', 'Almacén Central');

INSERT INTO TiendaOnline.DetalleEnvio (nro_envio, id_venta,
tipo_envio)
VALUES ((SELECT MAX(nro_envio) FROM Envio), 3, 'Domicilio'),
((SELECT MAX(nro_envio) FROM Envio), 5, 'Punto entrega');

-- Cancelar una compra
UPDATE TiendaOnline.Venta
SET estado = 'Cancelada'
WHERE id_venta = 1;

-- Cancelar un envío
UPDATE TiendaOnline.Envio
SET estado_envio = 'Cancelado'
WHERE nro_envio = 5;

-- Ver reputaciones (Elegimos el primero de c/u como ejemplo)

SELECT nombre, reputacion AS reputacionVendedor
FROM UsuarioRol r, Usuario u
WHERE r.id_usuario = u.id_usuario
AND rol= 'Vendedor'

```

```
LIMIT 1;
```

```
SELECT nombre, reputacion AS reputacionComprador  
FROM UsuarioRol r, Usuario u  
WHERE r.id_usuario = u.id_usuario  
AND rol= 'Comprador'  
LIMIT 1;
```

```
-- Contabilizar la cantidad de ventas de la plataforma para un  
rango de fechas
```

```
SELECT COUNT(*) AS cantidadVentas  
FROM TiendaOnline.Venta  
WHERE fecha BETWEEN '2024-01-01' AND '2024-12-31';
```

```
-- Contabilizar el valor total de ventas de la plataforma para un  
rango de fechas
```

```
SELECT SUM(valor) AS valorTotalVentas  
FROM (  
SELECT cantidad * precioFacturado AS valor  
FROM TiendaOnline.Detalle d, Venta v  
WHERE d.id_venta = v.id_venta  
AND fecha BETWEEN '2024-01-01' AND '2024-12-31'  
AND v.estado = 'Concretada') AS t;
```