

CASO DE ESTUDIO INTEGRADOR
CONSTRUCCIÓN DE UNA BASE DE DATOS PARTIENDO DE SUS DISEÑOS

ASIGNATURA
BASE DE DATOS 2

DOCENTE
JOHN JAIRO SEDANO SEGURA

PRESENTADO POR
JHONATTAN RESTREPO CASAÑAS
CRISTIAN CAMILO RAMÍREZ FLOREZ
JUAN PABLO NUÑEZ DUQUE
LAURA YASURYS COSSIO MORENO

POLITÉCNICO INTERNACIONAL

2025

Tabla de contenido

1	Construcción de Bases de Datos para ELECTROHUB.....	3
1.1	Descripción del escenario.	3
1.2	Construcción de los modelos que componen la Base de Datos	3
1.2.1	Modelo conceptual:.....	3
1.2.2	Modelo de entidad relacional:.....	6
1.2.3	Modelo lógico:	7
1.3	Implementación de la Base de Datos- <i>transact SQL</i>	8
1.4	Procedimientos almacenados-CRUD.....	13
1.5	Conclusiones:	14

1 Construcción de Bases de Datos para ELECTROHUB

1.1 Descripción del escenario.

El proyecto se basa en el diseño de una base de datos relacional para un sistema de comercio electrónico llamado **ElectroHub**, una tienda virtual especializada en la venta de electrodomésticos y artículos para el hogar. Este escenario fue elegido por su amplitud y aplicabilidad en la vida real, ya que permite modelar diversos procesos comerciales y administrativos mediante múltiples entidades y relaciones.

La base de datos tiene como objetivo gestionar eficientemente los diferentes aspectos del funcionamiento de la tienda en línea, incluyendo la administración de productos, categorías, proveedores, clientes, pedidos, pagos, envíos, promociones, reseñas e inventario. A través de esta estructura, se busca optimizar el manejo de la información, mejorar la experiencia del cliente y facilitar el crecimiento del negocio.

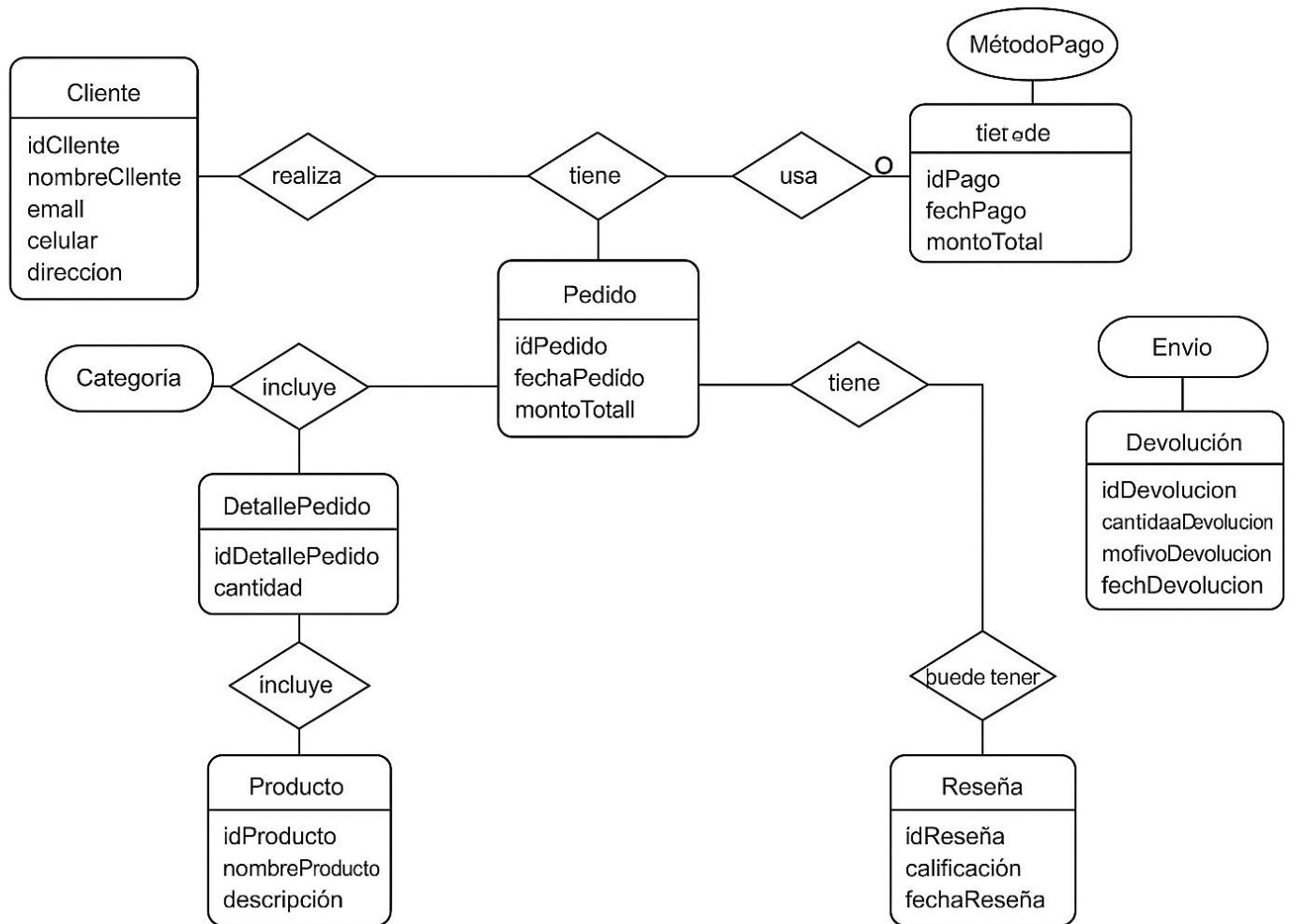
Entre las entidades principales se encuentran productos, clientes, pedidos y detalles de pedido, conectadas mediante relaciones uno a muchos y muchos a muchos, según el caso. Esta organización permitirá un seguimiento preciso del inventario, el historial de compras, la disponibilidad de productos y la trazabilidad de cada transacción, contribuyendo a una operación ordenada y escalable.

1.2 Construcción de los modelos que componen la Base de Datos

1.2.1 Modelo conceptual:

El siguiente modelo conceptual representa la estructura lógica de la base de datos para ElectroHub, una tienda en línea especializada en la venta de electrodomésticos y artículos para el hogar. A través del diagrama Entidad-Relación (ER), se identifican las entidades clave del sistema, sus atributos principales y las relaciones que existen entre ellas. Este modelo busca organizar de forma eficiente la información relacionada con los clientes, productos,

pedidos, pagos, devoluciones y reseñas, garantizando así una base sólida para el desarrollo e implementación de una base de datos relacional robusta y funcional.

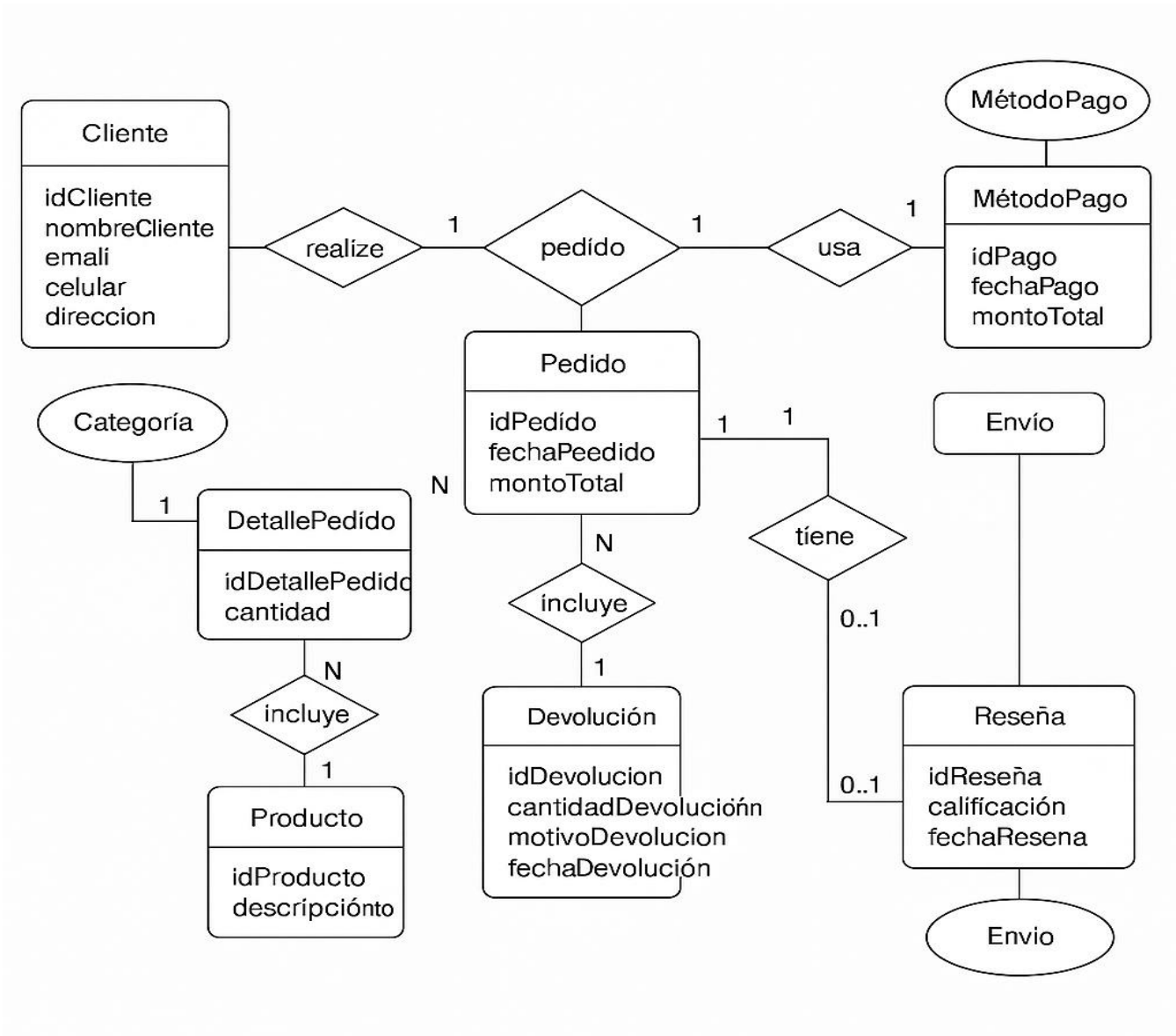


El sistema de comercio electrónico contempla las siguientes entidades principales y sus relaciones:

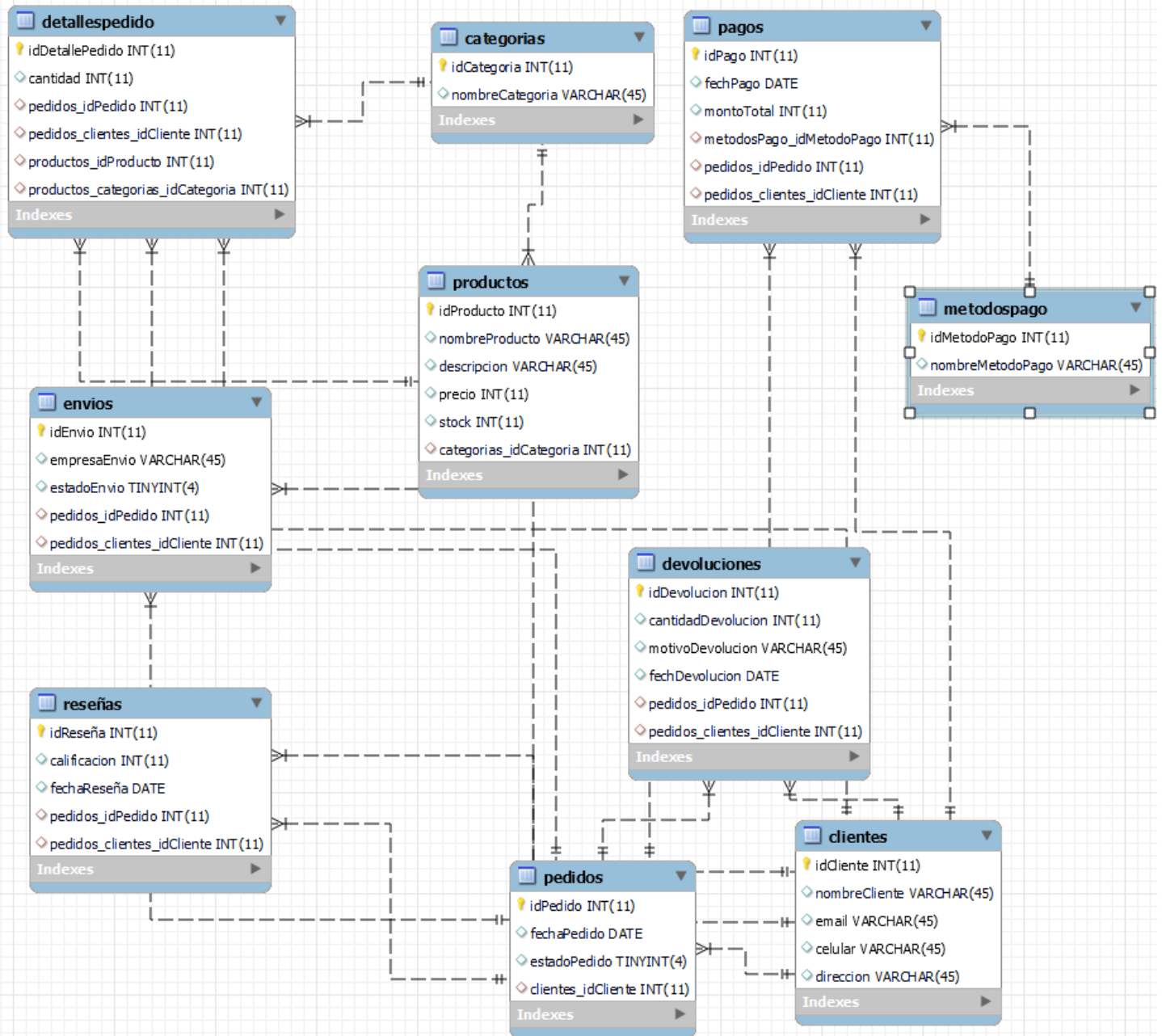
1. **Cliente:** contiene los datos personales del comprador, como idCliente, nombreCliente, email, celular y dirección.
 - Un cliente **realiza** uno o más **pedidos**.

2. **Pedido:** registra cada compra efectuada, con atributos como idPedido, fechaPedido y montoTotal.
 - Un pedido **tiene** uno o varios **detalles de pedido**.
 - Un pedido **usa** un **método de pago**.
 - Un pedido **tiene** una **reseña** opcional.
 - Un pedido puede generar un **envío**.
3. **DetallePedido:** representa los productos comprados en un pedido, con idDetallePedido y cantidad.
 - Un detalle de pedido **incluye** un **producto**.
 - Un producto **pertenece** a una **categoría**.
4. **Producto:** contiene la información de los artículos, como idProducto, nombreProducto y descripción.
5. **Categoría:** clasifica los productos del sistema.
6. **Pago:** define la forma de pago del pedido (idPago, fechaPago, montoTotal).
 - Un método de pago **se asocia** con uno o más **pedidos**.
7. **Reseña:** permite a los clientes valorar un pedido, con atributos como idReseña, calificación y fechaReseña.
 - Una reseña está **asociada** a un **pedido**.
8. **Envío:** proceso logístico de entrega de pedidos.
 - Un envío puede **generar** una **devolución**.
9. **Devolución:** registra datos como idDevolución, cantidadDevolución, motivoDevolución y fechaDevolución.
10. **Metodo de Pago:** clasifica los tipos de pago del sistema.

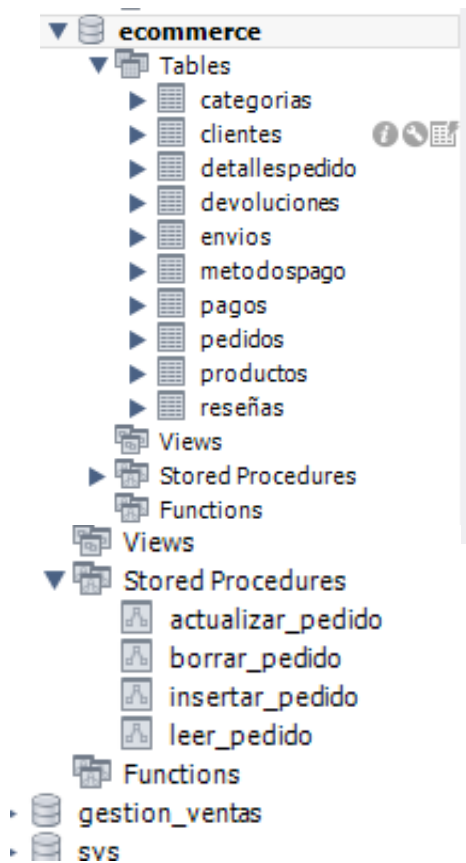
1.2.2 Modelo de entidad relacional:



1.2.3 Modelo lógico:



1.3 Implementación de la Base de Datos-*transact SQL*



```
1  -- Crear la base de datos ecommerce (si no existe)
2  • CREATE DATABASE IF NOT EXISTS ecommerce;
3
4  -- Usar la base de datos ecommerce
5  • USE ecommerce;
6
7  -- Tabla clientes
8  • CREATE TABLE clientes (
9      idCliente INT PRIMARY KEY,
10     nombreCliente VARCHAR(45),
11     email VARCHAR(45),
12     celular VARCHAR(45),
13     direccion VARCHAR(45)
14 );
15
16 -- Tabla metodosPago
17 • CREATE TABLE metodosPago (
18     idMetodoPago INT PRIMARY KEY,
19     nombreMetodoPago VARCHAR(45)
20 );
21
22 -- Tabla categorias
23 • CREATE TABLE categorias (
24     idCategoria INT PRIMARY KEY,
25     nombreCategoria VARCHAR(45)
26 );
27
```



```

28      -- Tabla productos
29  ● ○ CREATE TABLE productos (
30      idProducto INT PRIMARY KEY,
31      nombreProducto VARCHAR(45),
32      descripcion VARCHAR(45),
33      precio INT,
34      stock INT,
35      categorias_idCategoria INT,
36      FOREIGN KEY (categorias_idCategoria) REFERENCES categorias(idCategoria)
37  );
38
39      -- Tabla pedidos
40  ● ○ CREATE TABLE pedidos (
41      idPedido INT PRIMARY KEY AUTO_INCREMENT,
42      fechaPedido DATE,
43      estadoPedido TINYINT,
44      clientes_idCliente INT,
45      FOREIGN KEY (clientes_idCliente) REFERENCES clientes(idCliente)
46  );
47
48      -- Tabla pagos
49  ● ○ CREATE TABLE pagos (
50      idPago INT PRIMARY KEY,
51      fechPago DATE,
52      montoTotal INT,
53      metodosPago_idMetodoPago INT,
54      pedidos_idPedido INT,
55      pedidos_clientes_idCliente INT,
56      FOREIGN KEY (metodosPago_idMetodoPago) REFERENCES metodosPago(idMetodoPago),
57      FOREIGN KEY (pedidos_idPedido) REFERENCES pedidos(idPedido),
58      FOREIGN KEY (pedidos_clientes_idCliente) REFERENCES clientes(idCliente)
59  );
60
61      -- Tabla detallesPedido
62  ● ○ CREATE TABLE detallesPedido (
63      idDetallePedido INT PRIMARY KEY AUTO_INCREMENT,
64      cantidad INT,
65      pedidos_idPedido INT,
66      pedidos_clientes_idCliente INT,
67      productos_idProducto INT,
68      productos_categorias_idCategoria INT,
69      FOREIGN KEY (pedidos_idPedido) REFERENCES pedidos(idPedido),
70      FOREIGN KEY (pedidos_clientes_idCliente) REFERENCES clientes(idCliente),
71      FOREIGN KEY (productos_idProducto) REFERENCES productos(idProducto),
72      FOREIGN KEY (productos_categorias_idCategoria) REFERENCES categorias(idCategoria)
73  );
74

```

```

74
75 -- Tabla envios
76 • ⊖ CREATE TABLE envios (
77     idEnvio INT PRIMARY KEY,
78     empresaEnvio VARCHAR(45),
79     estadoEnvio TINYINT,
80     pedidos_idPedido INT,
81     pedidos_clientes_idCliente INT,
82     FOREIGN KEY (pedidos_idPedido) REFERENCES pedidos(idPedido),
83     FOREIGN KEY (pedidos_clientes_idCliente) REFERENCES clientes(idCliente)
84 );
85
86 -- Tabla devoluciones
87 • ⊖ CREATE TABLE devoluciones (
88     idDevolucion INT PRIMARY KEY,
89     cantidadDevolucion INT,
90     motivoDevolucion VARCHAR(45),
91     fechDevolucion DATE,
92     pedidos_idPedido INT,
93     pedidos_clientes_idCliente INT,
94     FOREIGN KEY (pedidos_idPedido) REFERENCES pedidos(idPedido),
95     FOREIGN KEY (pedidos_clientes_idCliente) REFERENCES clientes(idCliente)
96 );
97
98 -- Tabla reseñas
99 • ⊖ CREATE TABLE reseñas (
100     idReseña INT PRIMARY KEY,
101     calificacion INT,
102     fechaReseña DATE,
103     pedidos_idPedido INT,
104     pedidos_clientes_idCliente INT,
105     FOREIGN KEY (pedidos_idPedido) REFERENCES pedidos(idPedido),
106     FOREIGN KEY (pedidos_clientes_idCliente) REFERENCES clientes(idCliente)
107 );

```

```

108 -- Tabla clientes
109 • INSERT INTO clientes (idCliente, nombreCliente, email, celular, direccion) VALUES
110 (1, 'Laura Vargas', 'laura.vargas@tienda.com', '3101234567', 'Cr 10 # 5 - 20'),
111 (2, 'Pedro Jiménez', 'pedro.jimenez@tienda.com', '3169876543', 'Cl 25 # 12 - 45'),
112 (3, 'Carolina Ruiz', 'carolina.ruiz@tienda.com', '3225551111', 'Av 3 # 18 - 10'),
113 (4, 'Javier Torres', 'javier.torres@tienda.com', '3052229999', 'Dg 15 # 8 - 30'),
114 (5, 'Valentina Díaz', 'valentina.diaz@tienda.com', '3197772222', 'Tr 5 # 22 - 15');
115
116 -- Tabla metodosPago
117 • INSERT INTO metodosPago (idMetodoPago, nombreMetodoPago) VALUES
118 (1, 'Contra Entrega'),
119 (2, 'PayPal'),
120 (3, 'Nequi'),
121 (4, 'Daviplata'),
122 (5, 'Tarjeta de Crédito VISA');
123
124 -- Tabla categorias
125 • INSERT INTO categorias (idCategoria, nombreCategoria) VALUES
126 (1, 'Electrónicos'),
127 (2, 'Moda'),
128 (3, 'Hogar y Decoración'),
129 (4, 'Deportes y Aire Libre'),
130 (5, 'Libros y Música');
131
132 -- Tabla productos
133 • INSERT INTO productos (idProducto, nombreProducto, descripcion, precio, stock, categorias_idCategoria) VALUES
134 (1, 'Smartwatch Deportivo', 'Reloj inteligente con GPS', 250000, 25, 1),
135 (2, 'Camiseta Estampada', 'Camiseta de algodón con diseño moderno', 45000, 60, 2),
136 (3, 'Lámpara de Mesa LED', 'Lámpara con luz cálida regulable', 80000, 30, 3),
137 (4, 'Bicicleta Montaña', 'Bicicleta de montaña con 21 velocidades', 450000, 15, 4),
138 (5, 'Novela de Ciencia Ficción', 'Libro de ciencia ficción bestseller', 35000, 40, 5);
139
140 -- Tabla pedidos
141 • INSERT INTO pedidos (fechaPedido, estadoPedido, clientes_idCliente) VALUES
142 ('2025-04-28', 1, 1),
143 ('2025-04-27', 2, 3),
144 ('2025-04-26', 1, 2),
145 ('2025-04-25', 3, 5),
146 ('2025-04-24', 1, 4);
147
148 -- Tabla pagos
149 • INSERT INTO pagos (idPago, fechPago, montoTotal, metodosPago_idMetodoPago, pedidos_idPedido, pedidos_clientes_idCliente) VALUES
150 (1, '2025-04-28', 250000, 2, 1, 1),
151 (2, '2025-04-27', 90000, 3, 2, 3),
152 (3, '2025-04-26', 80000, 1, 3, 2),
153 (4, '2025-04-25', 450000, 4, 4, 5),
154 (5, '2025-04-24', 35000, 5, 5, 4);
155

```

```

156 -- Tabla detallesPedido
157 • INSERT INTO detallesPedido (cantidad, pedidos_idPedido, pedidos_clientes_idCliente, productos_idProducto, productos_categorias_idCategori
158 (1, 1, 1, 1, 1),
159 (2, 2, 3, 2, 2),
160 (1, 3, 2, 3, 3),
161 (1, 4, 5, 4, 4),
162 (1, 5, 4, 5, 5);
163
164 -- Tabla envios
165 • INSERT INTO envios (idEnvio, empresaEnvio, estadoEnvio, pedidos_idPedido, pedidos_clientes_idCliente) VALUES
166 (1, 'Envíos Rápidos S.A.S', 1, 1, 1),
167 (2, 'Logística Express', 2, 2, 3),
168 (3, 'Servicio al Cliente Ya', 1, 3, 2),
169 (4, 'Entrega Segura', 3, 4, 5),
170 (5, 'Tu Paquete a Tiempo', 1, 5, 4);
171
172 -- Tabla devoluciones
173 • INSERT INTO devoluciones (idDevolucion, cantidadDevolucion, motivoDevolucion, fechDevolucion, pedidos_idPedido, pedidos_clientes_idClient
174 (1, 1, 'No era el producto esperado', '2025-04-29', 1, 1),
175 (2, 1, 'Talla incorrecta', '2025-04-28', 2, 3),
176 (3, 1, 'Producto dañado durante el envío', '2025-04-30', 3, 2),
177 (4, 1, 'Color diferente al solicitado', '2025-05-01', 4, 5),
178 (5, 1, 'Pedido duplicado', '2025-05-02', 5, 4);
179
180 -- Tabla reseñas
181 • INSERT INTO reseñas (idReseña, calificacion, fechaReseña, pedidos_idPedido, pedidos_clientes_idCliente) VALUES
182 (1, 5, '2025-04-29', 1, 1),
183 (2, 4, '2025-04-28', 2, 3),
184 (3, 3, '2025-04-30', 3, 2),
185 (4, 5, '2025-05-01', 4, 5),
186 (5, 4, '2025-05-02', 5, 4);
187

```

1.4 Procedimientos almacenados-CRUD

Para demostrar la funcionalidad del sistema, se implementaron las operaciones CRUD sobre la tabla de Pedidos. Estas acciones permiten registrar nuevas órdenes de compra, obtener la información de pedidos realizados, modificar detalles cuando sea necesario y eliminar registros si corresponde. Con ello se garantiza una gestión completa y eficiente de los pedidos dentro del e-commerce ElectroHub.

```
1  -- Crear pedido (insertar un nuevo registro)
2  DELIMITER //
3  • CREATE PROCEDURE crear_pedido(IN p_fechaPedido DATE, IN p_estadoPedido TINYINT, IN p_clientes_idCliente INT)
4  BEGIN
5      INSERT INTO pedidos (fechaPedido, estadoPedido, clientes_idCliente)
6          VALUES (p_fechaPedido, p_estadoPedido, p_clientes_idCliente);
7  END //
8  DELIMITER ;
9
10 -- Obtener pedido por ID (aunque no lo solicitaste explícitamente, es útil para verificar)
11 DELIMITER //
12 • CREATE PROCEDURE obtener_pedido(IN p_idPedido INT)
13 BEGIN
14     SELECT * FROM pedidos WHERE idPedido = p_idPedido;
15 END //
16 DELIMITER ;
17
18 -- Actualizar pedido
19 DELIMITER //
20 • CREATE PROCEDURE actualizar_pedido(IN p_idPedido INT, IN p_fechaPedido DATE, IN p_estadoPedido TINYINT, IN p_clientes_idCliente INT)
21 BEGIN
22     UPDATE pedidos
23     SET fechaPedido = p_fechaPedido,
24         estadoPedido = p_estadoPedido,
25         clientes_idCliente = p_clientes_idCliente
26     WHERE idPedido = p_idPedido;
27 END //
28 DELIMITER ;
29
30 -- Borrar pedido (eliminar un registro)
31 DELIMITER //
32 • CREATE PROCEDURE borrar_pedido(IN p_idPedido INT)
33 BEGIN
34     DELETE FROM pedidos WHERE idPedido = p_idPedido;
35 END //
36 DELIMITER ;
37
```

1.5 Conclusiones:

- **Comprensión estructural del negocio:** El desarrollo del modelo conceptual permitió identificar y estructurar claramente los elementos esenciales que intervienen en el funcionamiento de un sistema de comercio electrónico, como clientes, productos, pedidos, métodos de pago y envíos.
- **Relaciones clave bien definidas:** Se establecieron las relaciones entre entidades de forma lógica, permitiendo representar cómo interactúan los usuarios con los pedidos, cómo se gestionan los productos y cómo se llevan a cabo las transacciones.
- **Base para el modelo lógico y físico:** Este modelo conceptual sirve como punto de partida sólido para construir posteriormente el modelo lógico (tablas, campos, claves primarias/foráneas) y el modelo físico (implementación en una base de datos real).
- **Escalabilidad y mantenimiento:** El modelo está diseñado de forma normalizada, lo que facilitará su mantenimiento, crecimiento y adaptabilidad en el futuro si se desea incorporar nuevas funciones como promociones, carritos de compra o múltiples direcciones de envío.
- **Facilita la comunicación:** El uso de este modelo ayuda a que todos los involucrados una visión compartida del sistema, minimizando errores y mejorando la planificación del desarrollo.
- **Permite el Uso de Procedimientos Almacenados para Automatizar Procesos:** ayuda a encapsular la lógica de las operaciones sobre la base de datos (insertar, actualizar, eliminar, consultar). Y así reducir errores.
- **Ventajas del CRUD Completo:** al construir un CRUD mediante procedimientos almacenados, se logra un control total sobre los datos y además facilita la integración con aplicaciones externas (web, móviles o de escritorio) que solo necesitan llamar a los procedimientos para manipular los registros.