



Universidad  
Nacional  
**de Loja**

**FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS  
Y LOS RECURSOS NATURALES NO RENOVABLES**

**CARRERA DE INGENIERÍA EN CIENCIAS DE LA  
COMPUTACIÓN  
TERCER CICLO “A”**

**ASIGNATURA: Base de Datos**

**ELABORADO POR:**

Anderson Mateo Coello Jaramillo

Steven Alexander Jumbo Jaramillo

**Docente encargado: ING. René Guamán**

**FECHA DE ENTREGA: 13/11/2025**

**LOJA – ECUADOR**

## 1. PROBLEMA

**Título:** Sistema de Gestión de Pedidos y Logística de "NextGen Hardware".

**Descripción:** La empresa "NextGen Hardware" es una distribuidora mayorista de componentes de hardware con su centro logístico principal en Loja. Sirve a 10 sucursales minoristas afiliadas en la provincia (Loja, Catamayo, Cariamanga, Alamor, Macará).

**Proceso Central:** El sistema debe gestionar el inventario de productos en la bodega central. Cada producto se define por su código, nombre, descripción, marca, categoría, stock disponible y precio unitario. Se deben almacenar las marcas (ej. 'Intel', 'NVIDIA') y categorías (ej. 'Procesadores', 'Tarjetas Gráficas') de forma independiente para normalizar la información.

Las sucursales (definidas por su ID, nombre, ciudad, dirección y teléfono) realizan pedidos a la bodega central.

Cada pedido debe registrar:

1. Un número de pedido único.
2. La fecha y hora exacta en que se realizó.
3. La sucursal que realiza el pedido.
4. El empleado del centro logístico que procesó y atendió el pedido. (Necesitamos almacenar los datos de los empleados, como ID, nombre, apellido y puesto).
5. El método de pago (ej. 'Transferencia', 'Crédito', 'Efectivo').

Un pedido está compuesto por uno o más productos. Para cada producto dentro de un pedido (detalle del pedido), se debe almacenar:

1. La cantidad solicitada.
2. El precio unitario al momento de la venta (podría diferir del precio actual en inventario, por ofertas).

**Requerimientos de Análisis (Consultas):** La gerencia necesita responder a las siguientes preguntas para optimizar la logística y el inventario:

1. ¿Cuáles son los 5 productos más vendidos (en unidades) en el último trimestre?
2. ¿Qué sucursal (nombre y ciudad) ha generado el mayor monto total en compras?
3. Mostrar los productos del inventario que nunca se han vendido.
4. ¿Qué cantón (ciudad de la sucursal) demanda más unidades de "Tarjetas Gráficas" (Categoría específica)?

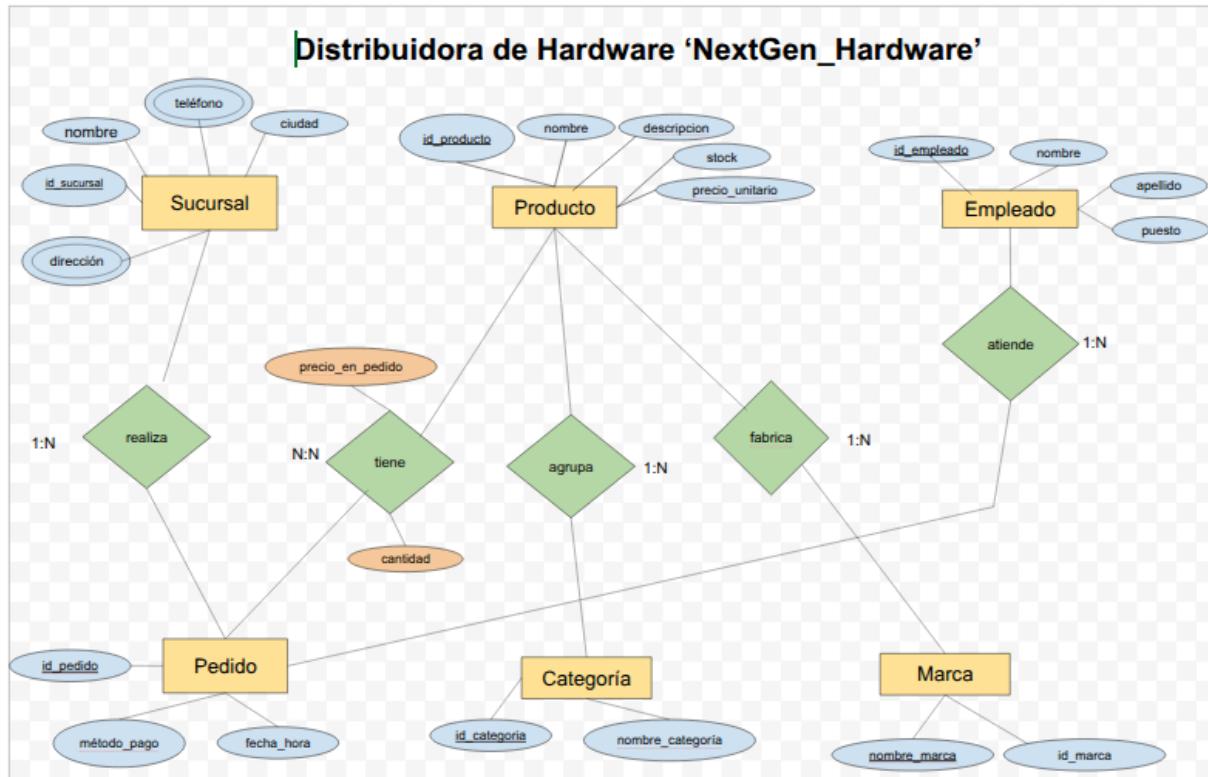
5. Listar todos los pedidos (ID de pedido y sucursal) atendidos por un empleado específico (ej. 'Juan Pérez').

**Link para acceder al GitHub con todos los archivos generados:**

<https://github.com/AndersonC15/examen-u1-bd-3ciclo.git>

## 2. MODELOS CONCEPTUALES

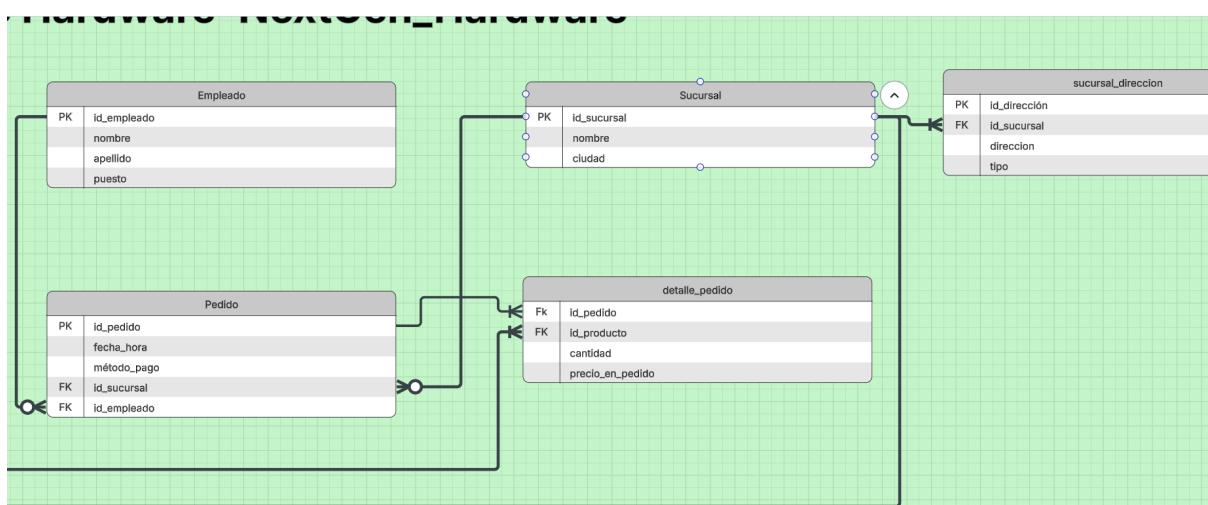
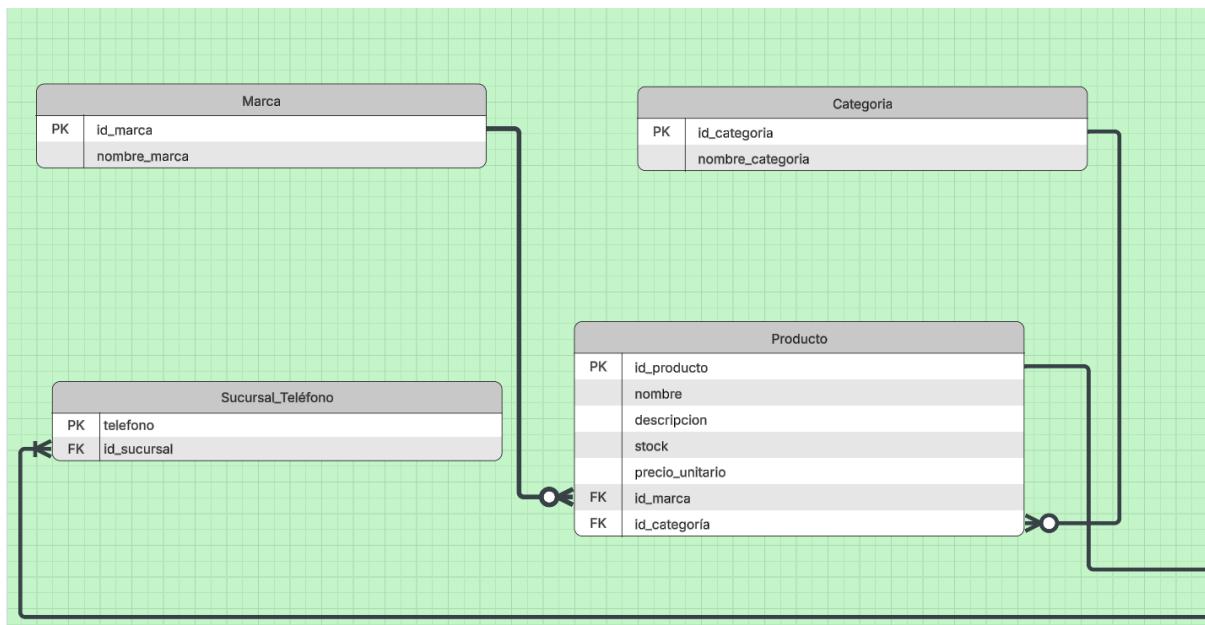
### 2.1 Modelo Entidad-Relación



**Link para acceder al Modelo Entidad-Relacion.**

[https://docs.google.com/drawings/d/1bQDuvjXHHW4zm8BUBSUsI4D8anvWqmf0rMPsQ43Y\\_c/edit?usp=sharing](https://docs.google.com/drawings/d/1bQDuvjXHHW4zm8BUBSUsI4D8anvWqmf0rMPsQ43Y_c/edit?usp=sharing)

### 2.2 Modelo Relacional



**Link para acceder al Modelo Relacional:**

[https://lucid.app/lucidchart/ad4538ac-5a66-48f8-9ccb-c0419a64b596/edit?viewport\\_loc=456%2C369%2C7281%2C3281%2C0\\_0&invitationId=inv\\_99d8242f-419e-4019-8a64-14da7c550cca](https://lucid.app/lucidchart/ad4538ac-5a66-48f8-9ccb-c0419a64b596/edit?viewport_loc=456%2C369%2C7281%2C3281%2C0_0&invitationId=inv_99d8242f-419e-4019-8a64-14da7c550cca)

### 3. GENERAR SQL

#### 3.1 Crear base de datos

CREATE DATABASE IF NOT EXISTS nextgen\_db

```
CHARACTER SET = 'utf8mb4'  
COLLATE = 'utf8mb4_general_ci';  
USE nextgen_db;
```

### 3.2 Crear tablas en la base de datos

---

```
CREATE TABLE categoria (  
    id_categoria INT PRIMARY KEY,  
    nombre_categoria VARCHAR(100) NOT NULL  
) ENGINE=InnoDB;
```

---

```
CREATE TABLE marca (  
    id_marca INT PRIMARY KEY,  
    nombre_marca VARCHAR(100) NOT NULL  
) ENGINE=InnoDB;
```

---

```
CREATE TABLE producto (  
    id_producto INT PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    descripcion TEXT,  
    stock INT,  
    precio_unitario DECIMAL(12,2),  
    id_marca INT,  
    id_categoria INT,
```

```
    CONSTRAINT fk_producto_marca FOREIGN KEY (id_marca) REFERENCES marca(id_marca),  
    CONSTRAINT fk_producto_categoria FOREIGN KEY (id_categoria) REFERENCES categoria(id_categoria)  
) ENGINE=InnoDB;
```

---

```
CREATE TABLE sucursal (  
    id_sucursal INT PRIMARY KEY,  
    nombre VARCHAR(255),  
    ciudad VARCHAR(100),  
    direccion VARCHAR(255),  
    telefono VARCHAR(50)  
) ENGINE=InnoDB;
```

---

```
CREATE TABLE empleado (  
    id_empleado INT PRIMARY KEY,  
    nombre VARCHAR(100),  
    apellido VARCHAR(100),  
    puesto VARCHAR(100)  
) ENGINE=InnoDB;
```

---

```
CREATE TABLE pedido (  
    id_pedido INT PRIMARY KEY,  
    fecha_hora DATETIME,  
    metodo_pago VARCHAR(50),  
    id_sucursal INT,
```

```
    id_empleado INT,  
    CONSTRAINT fk_pedido_sucursal FOREIGN KEY (id_sucursal) REFERENCES  
    sucursal(id_sucursal),  
    CONSTRAINT fk_pedido_empleado FOREIGN KEY (id_empleado) REFERENCES  
    empleado(id_empleado)  
) ENGINE=InnoDB;
```

---

```
CREATE TABLE pedido_producto (  
    id_pedido INT,  
    id_producto INT,  
    cantidad INT,  
    precio_en_pedido DECIMAL(12,2),  
    PRIMARY KEY (id_pedido, id_producto),  
    CONSTRAINT fk_pp_pedido FOREIGN KEY (id_pedido) REFERENCES  
    pedido(id_pedido),  
    CONSTRAINT fk_pp_producto FOREIGN KEY (id_producto) REFERENCES  
    producto(id_producto)  
) ENGINE=InnoDB;
```

---

### 3.3. Insertar datos en las tablas

---

```
INSERT INTO categoria (id_categoria, nombre_categoria) VALUES  
(1, 'Procesadores'),  
(2, 'Tarjetas Gráficas'),  
(3, 'Placas Madre'),  
(4, 'Memoria RAM'),
```

(5, 'Almacenamiento'),  
(6, 'Fuentes de Poder'),  
(7, 'Refrigeración'),  
(8, 'Gabinetes');

---

INSERT INTO marca (id\_marca, nombre\_marca) VALUES

(1,'Intel'),  
(2,'AMD'),  
(3,'NVIDIA'),  
(4,'ASUS'),  
(5,'Gigabyte'),  
(6,'Corsair'),  
(7,'Kingston'),  
(8,'Seagate'),  
(9,'Western Digital'),  
(10,'Cooler Master');

---

INSERT INTO producto  
(id\_producto,nombre,descripcion,stock,precio\_unitario,id\_marca,id\_categoria)  
VALUES

(1,'Almacenamiento Kingston 1','Almacenamiento de la marca Kingston modelo 1',10,777.34,7,5),  
(2,'Refrigeración ASUS 2','Refrigeración de la marca ASUS modelo 2',90,1420.71,4,7),  
(3,'Refrigeración AMD 3','Refrigeración de la marca AMD modelo 3',61,946.83,2,7);

**// Se agrega solo unos datos, ya que son demasiados.**

---

```
INSERT INTO sucursal (id_sucursal,nombre,ciudad,direccion,telefono) VALUES  
(1,'NextGen_LojaNorte','Loja','Av. Universitaria y Sucre','0991112233'),  
(2,'NextGen_LojaSur','Loja','Calle Rocafuerte y 9 de Octubre','0994445566'),  
(3,'NextGen_LojaCentro','Loja','Av. Amazonas y Colón','0985556677'),  
(4,'NextGen_LojaOeste','Loja','Av. Solano y Bolívar','0983332211'),  
(5,'NextGen_CatamayoCentro','Catamayo','Calle Bolívar y 10 de Agosto','0997778899'),  
(6,'NextGen_CatamayoNorte','Catamayo','Av. La Paz y Guayaquil','0998889900'),  
(7,'NextGen_Macara','Macara','Calle 24 de Mayo y Sucre','0999990011'),  
(8,'NextGen_Zozoranga','Zozoranga','Av. Principal y 5 de Junio','0991113322'),  
(9,'NextGen_Calvas','Calvas','Calle Central y Amazonas','0992224433'),  
(10,'NextGen_Paltas','Paltas','Av. Libertad y Colón','0993335544');
```

---

```
INSERT INTO empleado (id_empleado,nombre,apellido,puesto) VALUES  
(1,'Matteo','Coello','Vendedor'),  
(2,'Anderson','Pérez','Cajero'),  
(3,'Carla','Ramírez','Supervisor'),  
(4,'Daniel','Torres','Vendedor'),  
(5,'Lucía','Salinas','Cajero'),  
(6,'María','Lozano','Gerente'),  
(7,'Juan','Mendoza','Vendedor'),  
(8,'Ana','Silva','Cajero'),  
(9,'Roberto','Cárdenas','Supervisor'),  
(10,'Sofía','Guerrero','Vendedor');
```

---

```
INSERT INTO pedido  
(id_pedido,fecha_hora,metodo_pago,id_sucursal,id_empleado) VALUES  
(1,'2025-11-09 17:57:00','Efectivo',3,2),  
(2,'2025-11-06 15:44:00','Efectivo',5,5),  
(3,'2025-11-09 09:05:00','Efectivo',7,2),  
(4,'2025-11-03 14:53:00','Transferencia',8,5);  
// Se agrega solo unos datos, ya que son demasiados.
```

---

```
INSERT INTO pedido_producto (id_pedido,id_producto,cantidad,precio_en_pedido)  
VALUES  
(1,51,2,1181.23),  
(1,15,5,910.95),  
(1,27,4,147.03),  
(1,28,3,1482.66);  
// Se agrega solo unos datos, ya que son demasiados.
```

#### 4. CONSULTAS COMPLETAS - ÁLGEBRA RELACIONAL-SQL

##### Consulta 1 Álgebra Relacional

**Descripción:** Empleados que registraron pedidos pagados en efectivo entre 2025-11-13 00:00:00 y 2025-11-14 00:00:00.

$\pi_{\text{nombre}, \text{apellido}}(\sigma_{\text{metodo\_pago}='Efectivo' \text{ and } \text{fecha\_hora} \geq '2025-11-13 00:00:00' \text{ and } \text{fecha\_hora} < '2025-11-14 00:00:00'}(\text{pedido})) \bowtie \text{empleado}$

The screenshot shows a MySQL Workbench interface. At the top, there is a results grid with columns 'nombre' and 'apellido'. The data in the grid is:

nombre	apellido
Matteo	Coello
Maria	Lozano

Below the grid is a SQL editor window containing the following query:

```
SELECT DISTINCT e.id_empleado,
               e.nombre,
               e.apellido,
               e.puesto
  FROM pedido pe
 JOIN empleado e ON pe.id_empleado = e.id_empleado
 WHERE pe.metodo_pago = 'Efectivo'
   AND pe.fecha_hora >= '2025-11-13 00:00:00'
   AND pe.fecha_hora < '2025-11-14 00:00:00'
 ORDER BY e.id_empleado;
```

The SQL editor has several buttons at the bottom: 'Optimize', 'Undo optimize', 'Clear history', and 'Execute'.

## Consulta 1 en MySQL

```
SELECT DISTINCT e.id_empleado,
               e.nombre,
               e.apellido,
               e.puesto
  FROM pedido pe
 JOIN empleado e ON pe.id_empleado = e.id_empleado
 WHERE pe.metodo_pago = 'Efectivo'
   AND pe.fecha_hora >= '2025-11-13 00:00:00'
   AND pe.fecha_hora < '2025-11-14 00:00:00'
 ORDER BY e.id_empleado;
```

```
13 de nov 02:30 anderson-coello@Ubuntu2024lts: ~
anderson-coello@Ubuntu2024lts: ~
+-----+
| empleado | |
| marca    | |
| pedido   | |
| pedido_producto | |
| producto | |
| sucursal | |
+-----+
7 rows in set (0,00 sec)

mysql> -- Consulta B: empleados con pedidos en efectivo en la fecha 2025-11-13
mysql> SELECT DISTINCT e.id_empleado,
->                   e.nombre,
->                   e.apellido,
->                   e.puesto
-> FROM pedido pe
-> JOIN empleado e ON pe.id_empleado = e.id_empleado
-> WHERE pe.metodo_pago = 'Efectivo'
->       AND pe.fecha_hora >= '2025-11-13 00:00:00'
->       AND pe.fecha_hora < '2025-11-14 00:00:00'
-> ORDER BY e.id_empleado;
+-----+-----+-----+
| id_empleado | nombre | apellido | puesto |
+-----+-----+-----+-----+
|          1 | Matteo | Coello   | Vendedor |
|          6 | Maria  | Lozano  | Gerente  |
+-----+-----+-----+-----+
2 rows in set (0,00 sec)

mysql> |
```

## Consulta 2 Álgebra Relacional

**Descripción:** Muestra los nombres de los productos vendidos por empleados que trabajan en la sucursal con `id_sucursal == 3`.

`producto ⋙ pedido_producto ⋙ (σ id_sucursal==3(pedido))`

id_producto	nombre	descripcion	stock	precio_unitario	id_marca	id_cat
3	Refrigeración AMD 3	Refrigeración de la marca AMD modelo 3	61	946.83	2	7
15	Fuentes de Poder Cooler Master 15	Fuentes de Poder de la marca Cooler Master modelo 15	33	910.95	10	6
14	Almacenamiento NVIDIA 14	Almacenamiento de la marca NVIDIA modelo 14	11	1435.4	3	5
37	Almacenamiento NVIDIA 37	Almacenamiento de la marca NVIDIA modelo 37	54	951.88	3	5
26	Tarjetas Gráficas ASUS 26	Tarjetas Gráficas de la marca ASUS modelo 26	35	523.4	4	2
3	Refrigeración AMD 3	Refrigeración de la marca AMD modelo 3	61	946.83	2	7
27	Memoria RAM Corsair 27	Memoria RAM de la marca Corsair modelo 27	6	147.03	6	4
51	Memoria RAM Cooler Master 51	Memoria RAM de la marca Cooler Master modelo 51	78	1181.23	10	4
47	Fuentes de Poder Kingston 47	Fuentes de Poder de la marca Kingston modelo 47	76	920.9	7	6
24	Gabinetes Gigabyte 24	Gabinetes de la marca Gigabyte modelo 24	64	89.0	5	8
42	Tarjetas Gráficas Cooler Master 42	Tarjetas Gráficas de la marca Cooler Master modelo 42	65	264.09	10	2
26	Tarjetas Gráficas ASUS 26	Tarjetas Gráficas de la marca ASUS modelo 26	35	523.4	4	2
29	Memoria RAM AMD 29	Memoria RAM de la marca AMD modelo 29	96	146.85	2	4
60	Refrigeración Gigabyte 60	Refrigeración de la marca Gigabyte modelo 60	22	809.97	5	7
46	Memoria RAM Corsair 46	Memoria RAM de la marca Corsair modelo 46	57	174.24	6	4
59	Fuentes de Poder Cooler Master 59	Fuentes de Poder de la marca Cooler Master modelo 59	96	604.48	10	6
27	Memoria RAM Corsair 27	Memoria RAM de la marca Corsair modelo 27	6	147.03	6	4
31	Refrigeración Western Digital 31	Refrigeración de la marca Western Digital modelo 31	97	1096.67	9	7
16	Procesadores Intel 16	Procesadores de la marca Intel modelo 16	92	710.97	1	1
4	Almacenamiento Seagate 4	Almacenamiento de la marca Seagate modelo 4	9	248.16	8	5
27	Memoria RAM Corsair 27	Memoria RAM de la marca Corsair modelo 27	6	147.03	6	4
29	Memoria RAM AMD 29	Memoria RAM de la marca AMD modelo 29	96	146.85	2	4
47	Fuentes de Poder Kingston 47	Fuentes de Poder de la marca Kingston modelo 47	76	920.9	7	6
54	Refrigeración ASUS 54	Refrigeración de la marca ASUS modelo 54	78	1143.48	4	7

## Consulta 2 en MySQL

```

SELECT p.id_producto,
       p.nombre,
       p.descripcion,
       p.stock,
       SUM(pp.cantidad) AS total_vendido_en_sucursal
FROM pedido_producto pp
JOIN pedido pe ON pp.id_pedido = pe.id_pedido
JOIN producto p ON pp.id_producto = p.id_producto
WHERE pe.id_sucursal = 3
GROUP BY p.id_producto, p.nombre, p.descripcion, p.stock
ORDER BY total_vendido_en_sucursal DESC;

```

```

anderson-coello@Ubuntu2024:~ x anderson-coello@Ubuntu
-->      p.stock,
-->      SUM(pp.cantidad) AS total_vendido_en_sucursal
--> FROM pedido_producto pp
--> JOIN pedido pe ON pp.id_pedido = pe.id_pedido
--> JOIN producto p ON pp.id_producto = p.id_producto
--> WHERE pe.id_sucursal = 3
--> GROUP BY p.id_producto, p.nombre, p.descripcion, p.stock
--> ORDER BY total_vendido_en_sucursal DESC;
+-----+-----+-----+
| id_producto | nombre          | descripcion                | stock | total_vendido_en_sucursal |
+-----+-----+-----+
| 27 | Memoria RAM Corsair 27 | Memoria RAM de la marca Corsair modelo 27 | 6 | 11 |
| 4  | Almacenamiento Seagate 4 | Almacenamiento de la marca Seagate modelo 4 | 9 | 8 |
| 16 | Procesadores Intel 16 | Procesadores de la marca Intel modelo 16 | 92 | 6 |
| 29 | Memoria RAM AMD 29 | Memoria RAM de la marca AMD modelo 29 | 96 | 5 |
| 24 | Gabinetes Gigabyte 24 | Gabinetes de la marca Gigabyte modelo 24 | 64 | 5 |
| 47 | Fuentes de Poder Kingston 47 | Fuentes de Poder de la marca Kingston modelo 47 | 76 | 5 |
| 15 | Fuentes de Poder Cooler Master 15 | Fuentes de Poder de la marca Cooler Master modelo 15 | 33 | 5 |
| 11 | Fuentes de Poder Corsair 11 | Fuentes de Poder de la marca Corsair modelo 11 | 44 | 5 |
| 55 | Almacenamiento Intel 55 | Almacenamiento de la marca Intel modelo 55 | 37 | 5 |
| 37 | Almacenamiento NVIDIA 37 | Almacenamiento de la marca NVIDIA modelo 37 | 54 | 4 |
| 3  | Refrigeración AMD 3 | Refrigeración de la marca AMD modelo 3 | 61 | 4 |
| 7  | Fuentes de Poder NVIDIA 7 | Fuentes de Poder de la marca NVIDIA modelo 7 | 42 | 4 |
| 26 | Tarjetas Gráficas ASUS 26 | Tarjetas Gráficas de la marca ASUS modelo 26 | 35 | 4 |
| 31 | Refrigeración Western Digital 31 | Refrigeración de la marca Western Digital modelo 31 | 97 | 4 |
| 30 | Gabinetes Gigabyte 30 | Gabinetes de la marca Gigabyte modelo 30 | 79 | 3 |
| 28 | Fuentes de Poder Western Digital 28 | Fuentes de Poder de la marca Western Digital modelo 28 | 34 | 3 |
| 60 | Refrigeración Gigabyte 60 | Refrigeración de la marca Gigabyte modelo 60 | 22 | 3 |
| 23 | Gabinetes Gigabyte 23 | Gabinetes de la marca Gigabyte modelo 23 | 97 | 3 |
| 57 | Placas Madre Corsair 57 | Placas Madre de la marca Corsair modelo 57 | 43 | 2 |
| 14 | Almacenamiento NVIDIA 14 | Almacenamiento de la marca NVIDIA modelo 14 | 11 | 2 |
| 59 | Fuentes de Poder Cooler Master 59 | Fuentes de Poder de la marca Cooler Master modelo 59 | 96 | 2 |
| 51 | Memoria RAM Cooler Master 51 | Memoria RAM de la marca Cooler Master modelo 51 | 78 | 2 |
| 46 | Memoria RAM Corsair 46 | Memoria RAM de la marca Corsair modelo 46 | 57 | 2 |
| 42 | Tarjetas Gráficas Cooler Master 42 | Tarjetas Gráficas de la marca Cooler Master modelo 42 | 65 | 2 |
| 54 | Refrigeración ASUS 54 | Refrigeración de la marca ASUS modelo 54 | 78 | 2 |
+-----+-----+-----+
25 rows in set (0,00 sec)

mysql> 

```

## Consulta 3 Álgebra Relacional

**Descripción:** Muestra todos los pedidos realizados en el horario de la tarde durante el rango de fechas del 1 al 13 de noviembre de 2025.

$\sigma$  fecha\_hora $\geq$ '2025-11-01 12:00:00' and fecha\_hora $\leq$ '2025-11-13 18:00:00'

(pedido)

id_pedido	fecha_hora	metodo_pago	id_sucursal	id_empleado
14	2025-11-07 19:38:00	Transferencia	3	3
33	2025-11-07 09:07:00	Efectivo	5	5
19	2025-11-07 10:37:00	Transferencia	9	4
29	2025-11-04 13:59:00	Crédito	1	6
27	2025-11-09 12:27:00	Crédito	5	4
24	2025-11-13 11:23:00	Efectivo	1	6
7	2025-11-03 18:22:00	Efectivo	1	4
32	2025-11-04 20:50:00	Transferencia	9	3
31	2025-11-02 14:33:00	Transferencia	5	1
38	2025-11-02 13:17:00	Efectivo	8	4
15	2025-11-07 15:01:00	Transferencia	1	5
4	2025-11-03 14:53:00	Transferencia	8	5
9	2025-11-11 16:47:00	Transferencia	4	3
17	2025-11-04 14:43:00	Transferencia	3	1
3	2025-11-09 09:05:00	Efectivo	7	2
22	2025-11-09 10:50:00	Crédito	5	1
21	2025-11-02 14:12:00	Transferencia	3	3
18	2025-11-13 11:58:00	Crédito	8	2
20	2025-11-10 09:56:00	Crédito	5	3
23	2025-11-05 16:30:00	Efectivo	5	2
37	2025-11-11 08:24:00	Efectivo	5	4
2	2025-11-06 15:44:00	Efectivo	5	5
1	2025-11-09 17:57:00	Efectivo	3	2
30	2025-11-06 17:02:00	Efectivo	8	5
16	2025-11-05 15:59:00	Transferencia	3	6

(σ metodo\_pago=='Transferencia' and nombre\_marca=='ASUS')((marca ⋈ producto) ⋈ (pedido\_producto ⋈ pedido))  
 π nombre, nombre\_marca, descripcion, stock, metodo\_pago, precio\_unitario(σ metodo\_pago=='Transferencia' and nombre\_marca  
 σ fecha\_hora>='2025-11-01 12:00:00' and fecha\_hora<='2025-11-13 18:00:00' (pedido))

Optimize      Undo optimize      Clear history

σ fecha\_hora>='2025-11-01 12:00:00' and fecha\_hora<='2025-11-13 18:00:00' (pedido)

Execute

### Consulta 3 en MySQL

-- Consulta E1: pedidos en el rango temporal solicitado

```
SELECT id_pedido, fecha_hora, metodo_pago, id_sucursal, id_empleado
FROM pedido
WHERE fecha_hora >= '2025-11-01 12:00:00'
AND fecha_hora <= '2025-11-13 18:00:00'
ORDER BY fecha_hora;
```

```

anderson-coello@Ubuntu2024lts: ~
+-----+
-> FROM pedido
-> WHERE fecha_hora >= '2025-11-01 12:00:00'
-> AND fecha_hora <= '2025-11-13 18:00:00'
-> ORDER BY fecha_hora;
+-----+
| id_pedido | fecha_hora      | metodo_pago | id_sucursal | id_empleado |
+-----+
| 13        | 2025-11-01 16:26:00 | Crédito     | 7           | 6           |
| 38        | 2025-11-02 13:17:00 | Efectivo    | 8           | 4           |
| 21        | 2025-11-02 14:12:00 | Transferencia | 3           | 3           |
| 31        | 2025-11-02 14:33:00 | Transferencia | 5           | 1           |
| 12        | 2025-11-02 16:05:00 | Crédito     | 5           | 3           |
| 4         | 2025-11-03 14:53:00 | Transferencia | 8           | 5           |
| 7         | 2025-11-03 18:22:00 | Efectivo    | 1           | 4           |
| 29        | 2025-11-04 13:59:00 | Crédito     | 1           | 6           |
| 17        | 2025-11-04 14:43:00 | Transferencia | 3           | 1           |
| 32        | 2025-11-04 20:50:00 | Transferencia | 9           | 3           |
| 16        | 2025-11-05 15:59:00 | Transferencia | 3           | 6           |
| 23        | 2025-11-05 16:30:00 | Efectivo    | 5           | 2           |
| 2         | 2025-11-06 15:44:00 | Efectivo    | 5           | 5           |
| 30        | 2025-11-06 17:02:00 | Efectivo    | 8           | 5           |
| 35        | 2025-11-06 18:02:00 | Crédito     | 3           | 1           |
| 10        | 2025-11-06 20:51:00 | Crédito     | 6           | 1           |
| 33        | 2025-11-07 09:07:00 | Efectivo    | 5           | 5           |
| 19        | 2025-11-07 10:37:00 | Transferencia | 9           | 4           |
| 25        | 2025-11-07 12:30:00 | Crédito     | 10          | 5           |
| 15        | 2025-11-07 15:01:00 | Transferencia | 1           | 5           |
| 14        | 2025-11-07 19:38:00 | Transferencia | 3           | 3           |
| 36        | 2025-11-09 08:19:00 | Efectivo    | 3           | 1           |
| 3         | 2025-11-09 09:05:00 | Efectivo    | 7           | 2           |
| 22        | 2025-11-09 10:50:00 | Crédito     | 5           | 1           |
| 27        | 2025-11-09 12:27:00 | Crédito     | 5           | 4           |
| 1         | 2025-11-09 17:57:00 | Efectivo    | 3           | 2           |
| 20        | 2025-11-10 09:56:00 | Crédito     | 5           | 3           |
| 37        | 2025-11-11 08:24:00 | Efectivo    | 5           | 4           |
| 39        | 2025-11-11 09:25:00 | Efectivo    | 5           | 5           |
| 9         | 2025-11-11 16:47:00 | Transferencia | 4           | 3           |
| 34        | 2025-11-11 20:13:00 | Crédito     | 5           | 4           |
| 26        | 2025-11-12 10:24:00 | Crédito     | 3           | 6           |
| 28        | 2025-11-12 11:49:00 | Crédito     | 7           | 2           |
| 5         | 2025-11-12 19:15:00 | Efectivo    | 9           | 1           |
| 24        | 2025-11-13 11:23:00 | Efectivo    | 1           | 6           |
| 18        | 2025-11-13 11:58:00 | Crédito     | 8           | 2           |
| 6         | 2025-11-13 15:13:00 | Efectivo    | 10          | 6           |
| 8         | 2025-11-13 15:34:00 | Efectivo    | 2           | 1           |
+-----+
38 rows in set (0,00 sec)

mysql> 

```

## Consulta 4 Álgebra Relacional

$\pi_{\text{nombre, nombre_marca, descripcion, stock, metodo_pago, precio_unitario}}(\sigma_{\text{metodo_pago}=\text{'Transferencia'} \text{ and } \text{nombre_marca}=\text{'ASUS'}}((\text{marca} \bowtie \text{producto}) \bowtie (\text{pedido_producto} \bowtie \text{pedido})))$

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a results grid displaying two rows of data from a table. The columns are labeled: nombre, nombre\_marca, descripcion, stock, metodo\_pago, and precio\_unitario. The first row contains: Refrigeración ASUS 2, ASUS, Refrigeración de la marca ASUS modelo 2, 90, Transferencia, 1420.71. The second row contains: Tarjetas Gráficas ASUS 26, ASUS, Tarjetas Gráficas de la marca ASUS modelo 26, 35, Transferencia, 523.4.

nombre	nombre_marca	descripcion	stock	metodo_pago	precio_unitario
Refrigeración ASUS 2	ASUS	Refrigeración de la marca ASUS modelo 2	90	Transferencia	1420.71
Tarjetas Gráficas ASUS 26	ASUS	Tarjetas Gráficas de la marca ASUS modelo 26	35	Transferencia	523.4

In the main area, there is a query editor window containing the following SQL code:

```

 $\sigma$  fecha_hora>='2025-11-01 12:00:00' and fecha_hora<='2025-11-13 18:00:00' (pedido)
 $\sigma$  fecha_hora>='2025-11-01 12:00:00' and fecha_hora<='2025-11-13 18:00:00' (pedido)
 $\pi$  nombre, nombre_marca, descripcion, stock, metodo_pago, precio_unitario( $\sigma$  metodo_pago=='Transferencia' and nombre_marca=='ASUS')
    
```

Below the query editor are several buttons: Optimize, Undo optimize, Clear history, and Execute. The 'Execute' button is highlighted with a blue border.

## Consulta 4 en MySQL

**Descripcion:** Muestra nombre, nombre\_marca, descripcion, stock, metodo\_pago, precio\_unitario de la marca ASUS vendidos en pedidos con método de pago Transferencia.

SELECT

p.nombre,

m.nombre\_marca,

p.descripcion,

p.stock,

pe.metodo\_pago,

p.precio\_unitario

FROM marca m

JOIN producto p ON m.id\_marca = p.id\_marca

```

JOIN pedido_producto pp ON p.id_producto = pp.id_producto
JOIN pedido pe ON pp.id_pedido = pe.id_pedido
WHERE pe.metodo_pago = 'Transferencia'
AND m.nombre_marca = 'ASUS';

```

```

mysql> SELECT
->     p.nombre,
->     m.nombre_marca,
->     p.descripcion,
->     p.stock,
->     pe.metodo_pago,
->     p.precio_unitario
-> FROM marca m
-> JOIN producto p ON m.id_marca = p.id_marca
-> JOIN pedido_producto pp ON p.id_producto = pp.id_producto
-> JOIN pedido pe ON pp.id_pedido = pe.id_pedido
-> WHERE pe.metodo_pago = 'Transferencia'
->     AND m.nombre_marca = 'ASUS';
+-----+-----+-----+-----+-----+
| nombre | nombre_marca | descripcion | stock | metodo_pago | precio_unitario |
+-----+-----+-----+-----+-----+
| Refrigeración ASUS 2 | ASUS | Refrigeración de la marca ASUS modelo 2 | 90 | Transferencia | 1420.71 |
| Tarjetas Gráficas ASUS 26 | ASUS | Tarjetas Gráficas de la marca ASUS modelo 26 | 35 | Transferencia | 523.40 |
+-----+-----+-----+-----+-----+
2 rows in set (0,00 sec)

mysql>

```

## Consulta 5 Álgebra Relacional

**Descripción:** Muestra los productos que en algún pedido (`pedido_producto`) se vendieron a un precio (`precio_en_pedido`) mayor que su `precio_unitario` original (el que está en la tabla `producto`).

$\pi id\_producto, nombre, descripcion, precio\_unitario, precio\_en\_pedido, id\_pedido (\sigma precio\_en\_pedido > precio\_unitario (producto \bowtie pedido\_producto))$

The screenshot shows the MySQL Workbench interface. At the top, there's a dark header bar with the date "13 de nov 03:35" and a "Relational" tab. Below it is a toolbar with "Settings" and "Help" buttons. The main area is a query editor window with a table header row containing columns: id\_producto, nombre, descripcion, precio\_unitario, precio\_en\_pedido, and id\_pedido. The main text area contains a complex SQL query involving joins between 'pedido' and 'producto' tables, filtering by date and payment method. Below the query are buttons for "Optimize", "Undo optimize", "Clear history", and "Execute".

```
σ fecha_hora>='2025-11-01 12:00:00' and fecha_hora<='2025-11-13 18:00:00' (pedido)
n nombre, nombre_marca, descripcion, stock, metodo_pago, precio_unitario(σ metodo_pago=='Transferencia' and nombre_marca
n id_producto,nombre,descripcion,precio_unitario,precio_en_pedido,id_pedido(σ precio_en_pedido>precio_unitario(producto⊗pe
n,precio_unitario,precio_en_pedido,id_pedido(σ precio_en_pedido>precio_unitario(producto⊗pedido_producto)))|>
Optimize Undo optimize Clear history
n,precio_unitario,precio_en_pedido,id_pedido(σ precio_en_pedido>precio_unitario(producto⊗pedido_producto)))|> Execute
```

## Consulta 5 en MySQL

SELECT DISTINCT

```
p.id_producto,
p.nombre,
p.descripcion,
p.precio_unitario,
```

```
pp.precio_en_pedido,
pp.id_pedido
```

FROM pedido\_producto pp

JOIN producto p ON pp.id\_producto = p.id\_producto

WHERE pp.precio\_en\_pedido > p.precio\_unitario

```
ORDER BY p.id_producto, pp.id_pedido;
```

```
mysql> SELECT DISTINCT
->     p.id_producto,
->     p.nombre,
->     p.descripcion,
->     p.precio_unitario,
->     pp.precio_en_pedido,
->     pp.id_pedido
-> FROM pedido_producto pp
-> JOIN producto p ON pp.id_producto = p.id_producto
-> WHERE pp.precio_en_pedido > p.precio_unitario
-> ORDER BY p.id_producto, pp.id_pedido;
Empty set (0,00 sec)

mysql> █
```

## 5. VIEWS

### Primera VIEW

**Descripción:** Esta vista filtra todos los productos cuyo nivel de inventario (stock) es menor a 20 unidades, y los muestra junto con su marca y categoría.

```
CREATE VIEW vista_productos_bajo_stock AS
SELECT
    p.id_producto,
    p.nombre AS nombre_producto,
    p.stock,
    p.precio_unitario,
    m.nombre_marca,
    c.nombre_categoria
FROM producto p
JOIN marca m ON p.id_marca = m.id_marca
JOIN categoria c ON p.id_categoria = c.id_categoria
WHERE p.stock < 20;
```

### Ahora para ver la view

```
SELECT * FROM vista_productos_bajo_stock;
```

```

Empty set (0,00 sec)

mysql> CREATE VIEW vista_productos_bajo_stock AS
-> SELECT
->     p.id_producto,
->     p.nombre AS nombre_producto,
->     p.stock,
->     p.precio_unitario,
->     m.nombre_marca,
->     c.nombre_categoria
-> FROM producto p
-> JOIN marca m ON p.id_marca = m.id_marca
-> JOIN categoria c ON p.id_categoria = c.id_categoria
-> WHERE p.stock < 20;
Query OK, 0 rows affected (0,01 sec)

mysql> SELECT * FROM vista_productos_bajo_stock;
+-----+-----+-----+-----+-----+
| id_producto | nombre_producto | stock | precio_unitario | nombre_marca | nombre_categoria |
+-----+-----+-----+-----+-----+
|      25 | Tarjetas Gráficas AMD 25 |    5 |      1423.90 | AMD          | Tarjetas Gráficas |
|      9  | Placas Madre Corsair 9   |    7 |      973.78  | Corsair       | Placas Madre    |
|     27  | Memoria RAM Corsair 27  |    6 |      147.03  | Corsair       | Memoria RAM    |
|     58  | Memoria RAM Kingston 58 |    9 |      458.58  | Kingston     | Memoria RAM    |
|      1  | Almacenamiento Kingston 1 |   10 |      777.34  | Kingston     | Almacenamiento |
|      4  | Almacenamiento Seagate 4 |    9 |      248.16  | Seagate      | Almacenamiento |
|     14  | Almacenamiento NVIDIA 14 |   11 |      1435.40 | NVIDIA        | Almacenamiento |
|     19  | Almacenamiento NVIDIA 19 |   13 |      712.80  | NVIDIA        | Almacenamiento |
|     36  | Almacenamiento Kingston 36 |   18 |      713.12  | Kingston     | Almacenamiento |
+-----+-----+-----+-----+-----+
9 rows in set (0,00 sec)

mysql>

```

## Segunda View

**Descripción:** Esta vista muestra cuánto ha vendido cada empleado en total (sumando todas las ventas registradas por ellos en pedido y pedido\_producto).

```

CREATE VIEW vista_ventas_por_empleado AS
SELECT
    e.id_empleado,
    e.nombre AS nombre_empleado,
    e.apellido,
    SUM(pp.precio_en_pedido * pp.cantidad) AS total_facturado
FROM empleado e
JOIN pedido pe ON e.id_empleado = pe.id_empleado
JOIN pedido_producto pp ON pe.id_pedido = pp.id_pedido
GROUP BY e.id_empleado, e.nombre, e.apellido;

```

## Ahora para ver la view

```
SELECT * FROM vista_ventas_por_empleado ORDER BY total_facturado DESC;
```

```

mysql> CREATE VIEW vista_ventas_por_empleado AS
-> SELECT
->     e.id_empleado,
->     e.nombre AS nombre_empleado,
->     e.apellido,
->     SUM(pp.precio_en_pedido * pp.cantidad) AS total_facturado
-> FROM empleado e
-> JOIN pedido pe ON e.id_empleado = pe.id_empleado
-> JOIN pedido_producto pp ON pe.id_pedido = pp.id_pedido
-> GROUP BY e.id_empleado, e.nombre, e.apellido;
Query OK, 0 rows affected (0,01 sec)

mysql> SELECT * FROM vista_ventas_por_empleado ORDER BY total_facturado DESC;
+-----+-----+-----+
| id_empleado | nombre_empleado | apellido | total_facturado |
+-----+-----+-----+
|      2 | Anderson       | Pérez    |      70801.48 |
|      5 | Lucía          | Salinas  |      66412.47 |
|      4 | Daniel          | Torres   |      64395.83 |
|      1 | Matteo          | Coello   |      63614.13 |
|      3 | Carla           | Ramírez  |      52944.45 |
|      6 | María           | Lozano   |      42858.24 |
+-----+-----+-----+
6 rows in set (0,00 sec)

mysql> █

```

### Tercera View

**Descripción:** Esta vista combina toda la información relevante de los pedidos: qué se vendió, cuándo, en qué sucursal, quién lo vendió, a qué precio y cuánta cantidad.

```

CREATE VIEW vista_detalle_pedidos AS
SELECT
    pe.id_pedido,
    pe.fecha_hora,
    pe.metodo_pago,
    s.nombre AS sucursal,
    e.nombre AS empleado,
    e.apellido,
    p.nombre AS producto,
    pp.cantidad,
    pp.precio_en_pedido
FROM pedido pe
JOIN sucursal s ON pe.id_sucursal = s.id_sucursal
JOIN empleado e ON pe.id_empleado = e.id_empleado
JOIN pedido_producto pp ON pe.id_pedido = pp.id_pedido
JOIN producto p ON pp.id_producto = p.id_producto;

```

## Ahora para ver la view

```
SELECT * FROM vista_detalle_pedidos LIMIT 10;
```

```
mysql> CREATE VIEW vista_detalle_pedidos AS
-> SELECT
->     pe.id_pedido,
->     pe.fecha_hora,
->     pe.metodo_pago,
->     s.nombre AS sucursal,
->     e.nombre AS empleado,
->     e.apellido,
->     p.nombre AS producto,
->     pp.cantidad,
->     pp.precio_en_pedido
-> FROM pedido pe
-> JOIN sucursal s ON pe.id_sucursal = s.id_sucursal
-> JOIN empleado e ON pe.id_empleado = e.id_empleado
-> JOIN pedido_producto pp ON pe.id_pedido = pp.id_pedido
-> JOIN producto p ON pp.id_producto = p.id_producto;
Query OK, 0 rows affected (0,01 sec)

mysql> SELECT * FROM vista_detalle_pedidos LIMIT 10;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_pedido | fecha_hora | metodo_pago | sucursal | empleado | apellido | producto | cantidad | precio_en_pedido |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 7 | 2025-11-03 18:22:00 | Efectivo | NextGen_LojaNorte | Daniel | Torres | Memoria RAM Kingston 20 | 4 | 688.63 |
| 7 | 2025-11-03 18:22:00 | Efectivo | NextGen_LojaNorte | Daniel | Torres | Almacenamiento NVIDIA 37 | 5 | 951.88 |
| 7 | 2025-11-03 18:22:00 | Efectivo | NextGen_LojaNorte | Daniel | Torres | Almacenamiento Gigabyte 50 | 1 | 231.51 |
| 15 | 2025-11-07 15:01:00 | Transferencia | NextGen_LojaNorte | Lucia | Salinas | Refrigeración ASUS 2 | 5 | 1420.71 |
| 15 | 2025-11-07 15:01:00 | Transferencia | NextGen_LojaNorte | Lucia | Salinas | Almacenamiento NVIDIA 19 | 4 | 712.80 |
| 15 | 2025-11-07 15:01:00 | Transferencia | NextGen_LojaNorte | Lucia | Salinas | Memoria RAM Kingston 20 | 3 | 688.63 |
| 15 | 2025-11-07 15:01:00 | Transferencia | NextGen_LojaNorte | Lucia | Salinas | Almacenamiento Cooler Master 41 | 2 | 803.92 |
| 15 | 2025-11-07 15:01:00 | Transferencia | NextGen_LojaNorte | Lucia | Salinas | Gabinetes Kingston 44 | 4 | 373.16 |
| 24 | 2025-11-13 11:23:00 | Efectivo | NextGen_LojaNorte | Maria | Lozano | Memoria RAM Kingston 20 | 1 | 688.63 |
| 24 | 2025-11-13 11:23:00 | Efectivo | NextGen_LojaNorte | Maria | Lozano | Fuentes de Poder Seagate 35 | 1 | 680.64 |
+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0,00 sec)

mysql> |
```

## ANEXOS

