

# TECNOLOGICO NACIONAL DE MEXICO

## INSTITUTO TECNOLÓGICO SUPERIOR DE COSAMALOAPAN

### Nombre del trabajo:

Punto de venta de hardware

### Presenta (n):

Zacarias Ventura Martinez

Ángel Roberto Casoluengo Romero

### Asignatura:

Taller de base de datos

### Docente:

Roberto Zamudio Portilla

## Índice

<b>INTRODUCCION .....</b>	<b>1</b>
<b>IDENTIFICACION DE ENTIDADES ATRIBUTOS Y LLAVES PRIMARIAS .....</b>	<b>3</b>
<b>ANALISIS DE CARDINALIDAD .....</b>	<b>4</b>
<b>MODELO ENTIDAD RELACION .....</b>	<b>5</b>
<b>MODELO RELACIONAL.....</b>	<b>6</b>
<b>DICCIONARIO DE DATOS .....</b>	<b>7</b>
<b>SCRIPT DE LA BASE DE DATOS .....</b>	<b>9</b>
<b>CONSULTAS MULTICELANEAS .....</b>	<b>11</b>
<b>PROCEDIMIENTOS ALMACENADOS.....</b>	<b>20</b>
<b>FUNCIONES.....</b>	<b>24</b>
<b>DISPARADORES .....</b>	<b>24</b>
<b>USUARIOS .....</b>	<b>24</b>

## INTRODUCCION

En el siguiente proyecto a observar va orientado a la implementación de un sistema de ventas, en donde un usuario ya registrado podrá realizar ciertas acciones en el sistema mismo, como realizar ventas, hacer cortes de caja, consultar los productos existentes, consultar los clientes existentes, agregar, eliminar y actualizar los datos de todas las entidades existentes en el sistema.

Cabe destacar que tambien podrán visualizar ciertas consultas asignadas al sistema y por último una bitácora de los movimientos del usuario.

## PROBLEMÁTICA

La empresa de venta de hardware necesita una base de datos con la siguiente información: productos, ventas, proveedores, clientes y categorías.

Del **proveedor** se almacenará: ID, Nombre, RFC, Dirección, Teléfono y pagina Web.

De los **productos** se almacenará SKU, Nombre, Precio, PrecioVenta, Stock y Proveedor.

De las **ventas** se almacenará: Número factura, Fecha, Cliente y Monto final.

De los **clientes** se almacenará: ID, Nombre, Email, Dirección y Teléfono (Puede tener varios teléfonos de contacto). La dirección se compone con los siguientes datos (Ciudad, comuna, calle, numero).

Las **categorías** se componen de ID Categoría, Nombre y Descripción.

En una **categoría** se agruparán varios **productos**, pero un **producto** solo pertenecerá a una **categoría**. Un **proveedor** podrá abastecer varios **productos**, pero un **producto** será surtido por un solo **proveedor**. Una **venta** podrá incluir muchos productos y un **producto** solo se puede incluir en una venta. En una **venta** detallada se debe guardar cada producto que se lleva el cliente, la cantidad y precio al momento de la venta, considerando tambien su monto por cada producto. Una **venta** solo podrá ser afectada por un **cliente**, pero un **cliente** podrá actuar en varias **ventas**.

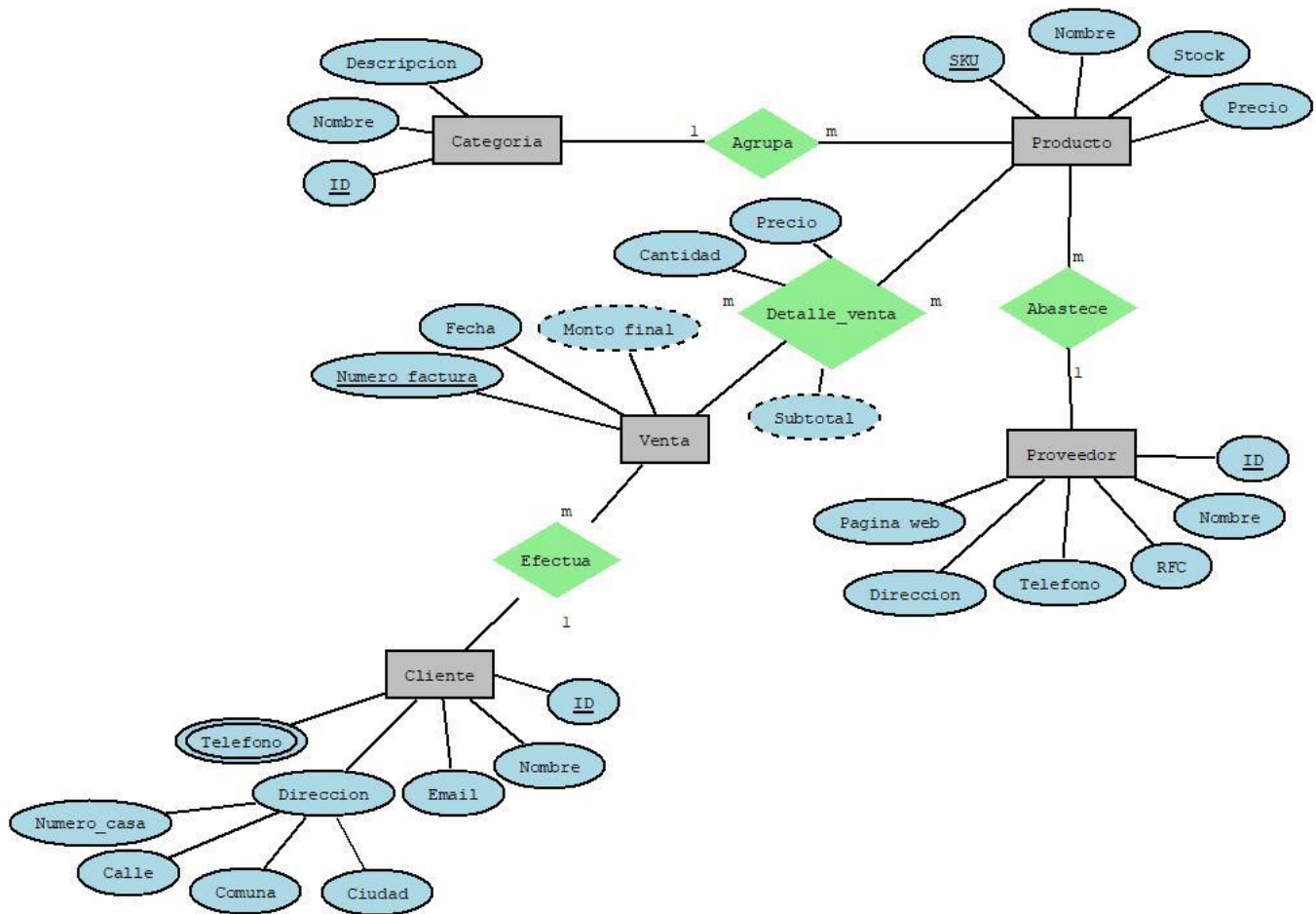
## IDENTIFICACION DE ENTIDADES ATRIBUTOS Y LLAVES PRIMARIAS

Entidad	Atributo	Llave primaria
<b>Proveedor</b>	ID, Nombre, RFC, Dirección, Teléfono y Web.	ID
<b>Producto</b>	SKU, Nombre, Precio, PrecioVenta Stock y Proveedor.	SKU.
<b>Venta</b>	Número de factura, Fecha, Cliente y Monto final.	Número de factura
<b>Cliente</b>	ID, Nombre, Email, Dirección y Teléfono.	ID
<b>Categoría</b>	ID, Nombre y Descripción.	ID

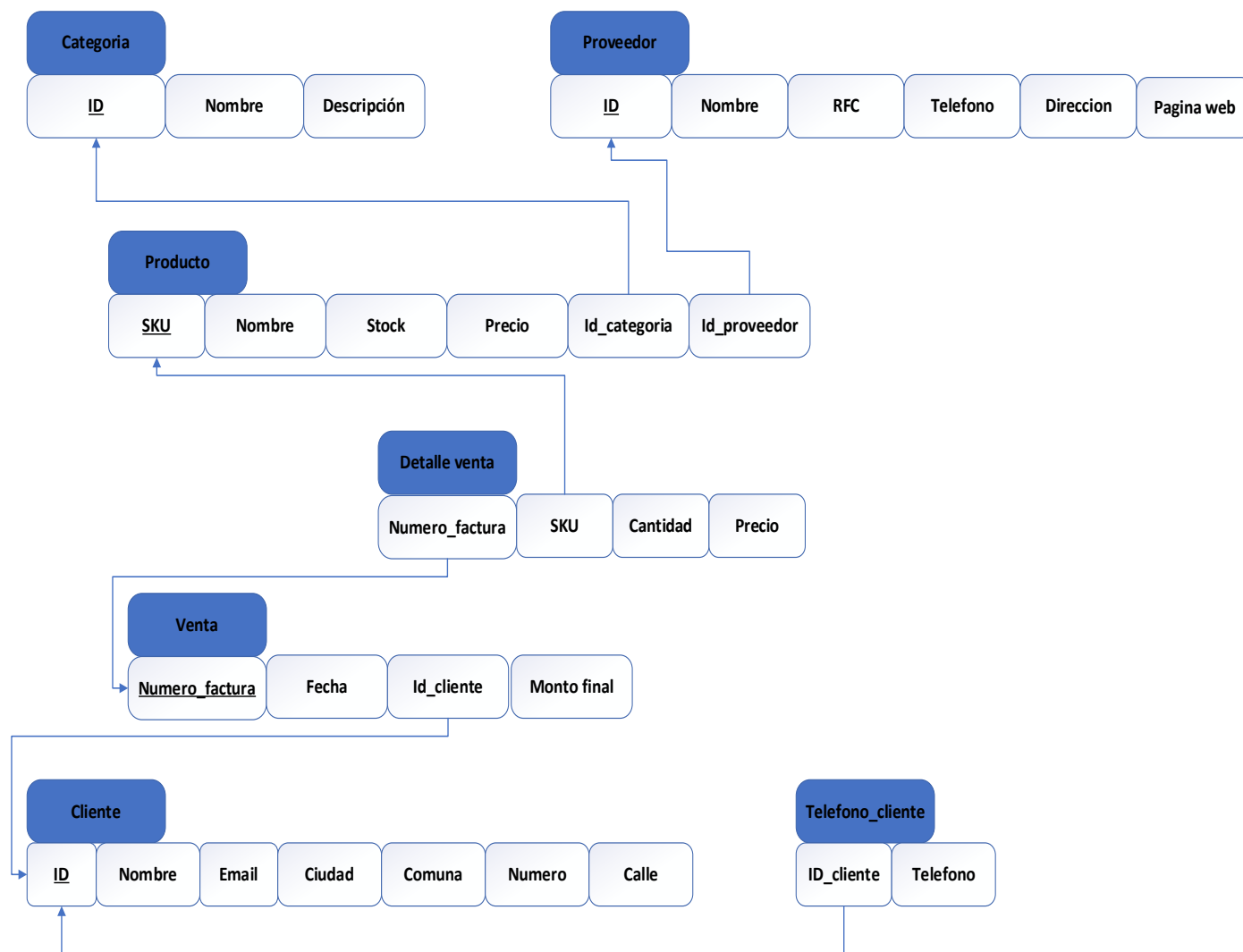
## ANÁLISIS DE CARDINALIDAD

Entidad A	Rotulo/Cardinalidad	Entidad B
Categoría (1, m)	Agrupar 1 : m	Producto (1, 1)
Proveedor (1, m)	Abastece 1 : m	Producto (1, 1)
Cliente (1, m)	Efectúa 1 : m	Venta (1, 1)
Venta (1, m)	Detalle venta m = m	Producto (1, m)

## MODELO ENTIDAD RELACION



## MODELO RELACIONAL





## DICCIONARIO DE DATOS

Tabla: Categoría

Campo	Tipo de dato	Longitud	PK	FK	Restricción
ID	Char	5	X		Not null
Nombre	Varchar	100			Not null
Descripción	Text				Not null

Tabla: Producto

Campo	Tipo de dato	Longitud	PK	FK	Restricción
SKU	Varchar	10	X		Not null
Nombre	Varchar	100			Not null
Stock	Int	Default 0			Not null
Precio	double	Default 0			Not null
Id_categoria	Int			X	Not null
Id_proveedor	Char	5		X	Not null

Tabla: Proveedor

Campo	Tipo de dato	Longitud	PK	FK	Restricción
ID	Char	5	X		Not null
Nombre	Varchar	100			Not null
RFC	Varchar	10			Not null
Teléfono	Char	10			Not null
Pagina web	Text				Not null

Tabla: Detalle\_venta

Campo	Tipo de dato	Longitud	PK	FK	Restricción
Numero_factura	Char	5		X	Not null
SKU	Varchar	10		X	Not null
Cantidad	Int	Default 0			Not null
Precio	Double				Not null

Tabla: Venta

Campo	Tipo de dato	Longitud	PK	FK	Restricción
Numero_factura	Char	5	X		Not null
Fecha	Date				Not null
Id_cliente	Char	5		X	Not null
Monto final	Double				Not null

Tabla: Cliente

Campo	Tipo de dato	Longitud	PK	FK	Restricción
ID	Char	5	X		Not null
Nombre	Varchar	80			Not null
Email	Text				Not null
Ciudad	Varchar	100			Not null
Comuna	Varchar	100			
Calle	Varchar	100			
Numero_casa	Int				

Tabla: Telefono\_cliente

Campo	Tipo de dato	Longitud	PK	FK	Restricción
ID_cliente	Char	5	X		Not null
Teléfono	Char	10			Not null

## SCRIPT DE LA BASE DE DATOS

En la siguiente sección creamos la base de datos y las tablas respectivas a nuestra problemática.

```
CREATE DATABASE VentaHardware;
```

```
CREATE TABLE users
```

```
(  
  ID CHAR(5) PRIMARY KEY,  
  Usuario VARCHAR(50) NOT NULL,  
  Rol_usuario ENUM('Administrador', 'Vendedor') NOT NULL,  
  Constraseña VARCHAR(30) NOT NULL,  
  Estatus ENUM('Activo', 'Inactivo')  
);
```

```
CREATE TABLE Categoria
```

```
(  
  ID INT PRIMARY KEY AUTO_INCREMENT,  
  Nombre VARCHAR(100) NOT NULL,  
  Descripcion TEXT NOT NULL,  
  Estatus ENUM('Activo', 'Inactivo')  
);
```

```
CREATE TABLE Producto
```

```
(  
  SKU VARCHAR(5) PRIMARY KEY,  
  Nombre VARCHAR(100) NOT NULL,  
  Stock INT DEFAULT 0 NOT NULL,  
  Precio DOUBLE NOT NULL,  
  Precio_Venta DOUBLE DEFAULT 0 NOT NULL,  
  ID_categoria INT DEFAULT 0 NOT NULL,  
  ID_proveedor CHAR(5) NOT NULL,  
  Estatus ENUM('Activo', 'Inactivo') NOT NULL,  
  CONSTRAINT FOREIGN KEY (ID_categoria) REFERENCES categoria (ID),  
  CONSTRAINT FOREIGN KEY (ID_proveedor) REFERENCES proveedor (ID)  
);
```

```
CREATE TABLE Proveedor
```

```
(  
  ID CHAR(5) PRIMARY KEY,  
  Nombre VARCHAR(100) NOT NULL,  
  RFC VARCHAR(10) NOT NULL,  
  Telefono CHAR(10) NOT NULL,  
  Direccion VARCHAR(100) NOT NULL,  
  Pagina_web TEXT NOT NULL,  
  Estatus ENUM('Activo', 'Inactivo')  
);
```

```
CREATE TABLE detalle_venta
```

```
(
  Num_factura INT NOT NULL AUTO_INCREMENT,
  SKU VARCHAR(10) NOT NULL,
  Cantidad INT DEFAULT 0 NOT NULL,
  Precio DOUBLE DEFAULT 0 NOT NULL,
  Importe DOUBLE DEFAULT 0 NOT NULL,
  CONSTRAINT FOREIGN KEY (Num_factura) REFERENCES venta (Numero_factura),
  CONSTRAINT FOREIGN KEY (SKU) REFERENCES producto (SKU)
);
```

CREATE TABLE Venta

```
(
  Numero_factura INT PRIMARY KEY AUTO_INCREMENT,
  Fecha DATE NOT NULL,
  ID_cliente CHAR(5) NOT NULL,
  Monto_final DOUBLE DEFAULT 0,
  CONSTRAINT FOREIGN KEY (ID_cliente) REFERENCES cliente (ID)
);
```

CREATE TABLE Telefono\_cliente

```
(
  ID_cliente CHAR(5) NOT NULL,
  Telefono CHAR(10) NOT NULL,
  CONSTRAINT FOREIGN KEY (ID_cliente) REFERENCES Cliente (ID)
);
```

CREATE TABLE corte\_caja

```
(
  N°_corte INT PRIMARY KEY AUTO_INCREMENT,
  Usuario VARCHAR(50),
  Fecha_corte DATETIME NOT NULL,
  Total_ventas INT DEFAULT 0,
  Total_Efectivo DOUBLE DEFAULT 0
);
```

CREATE TABLE bitacora

```
(
  Numero_bitacora INT PRIMARY KEY AUTO_INCREMENT,
  usuario VARCHAR(50) NOT NULL,
  Fecha_entrada DATETIME NOT NULL,
  Fecha_salida DATETIME NOT NULL
);
```

## CONSULTAS MULTICELANEAS

### Básicas

1. Mostrar el RFC y teléfono del proveedor que su página web es 'www.TecnoEUDIO.com'

```
SELECT RFC, Telefono FROM proveedor WHERE Pagina_web = 'www.TecnoEUDIO.com';
```

proveedor (1r × 2c)	
RFC	Telefono
RTXC153927	2881236745

2. Mostrar el nombre de los clientes que contengan 'unidades de estado solido'

```
SELECT Nombre FROM categoria WHERE Descripcion LIKE '%unidades de estado solido%';
```

categoria (1r × 1c)	
Nombre	
ALMACENAMIENTO	

3. Mostrar el nombre de los clientes que contengan 'hotmail' en su correo

```
SELECT nombre FROM cliente WHERE Email LIKE '%hotmail%';
```

cliente (2r × 1c)	
nombre	
grupoVentura	
Siemens	

4. Mostrar los productos que tienen un precio mayor a 5000

```
SELECT * FROM producto WHERE precio > 5000;
```

producto (4r × 6c)						
SKU	Nombre	Stock	Precio	ID_categoria	ID_proveedor	
09FGV	LENOVO	100	12,400	2	00342	
1DFFG	Iphone 12	10	16,000	4	39471	
2CS13	Asus	26	39,000	2	00342	
4HV0N	Samsung Galaxy s7	100	6,000	4	39471	

5. Mostrar todos los productos cuya categoría están dentro a '3' y '1'

```
SELECT * FROM producto WHERE Id_categoria IN ('3', '1');
```

producto (4r × 6c)						
SKU	Nombre	Stock	Precio	ID_categoria	ID_proveedor	
2R46G	JBL	80	3,100	1	32910	
3RT9V	Beat	100	1,500	1	32910	
67CVN	USB Kingston	70	150	3	00342	
J45DR	Audifonos Bose	30	1,400	1	32910	

6. Mostrar todos los productos cuya cantidad almacenada es menor de 100

```
SELECT * FROM producto WHERE Stock < 30;
```

producto (3r × 6c)						
SKU	Nombre	Stock	Precio	ID_categoria	ID_proveedor	
1DFFG	Iphone 12	10	16,000	4	39471	
2CS13	Asus	26	39,000	2	00342	
2VBNM	Xaomi Redmi Note 9	5	4,100	4	00342	

7. Mostrar los productos que no pertenecen a la categoría 1 y 2

```
SELECT * FROM producto WHERE Id_categoria NOT IN ('1', '2');
```

producto (4r × 6c)						
SKU	Nombre	Stock	Precio	ID_categoria	ID_proveedor	
1DFFG	Iphone 12	10	16,000	4	39471	
2VBNM	Xaomi Redmi Note 9	5	4,100	4	00342	
4HV0N	Samsung Galaxy s7	100	6,000	4	39471	
67CVN	USB Kingston	70	150	3	00342	

Con funciones de agregación

### 8. Mostrar la cantidad de categorías

```
SELECT COUNT(ID) AS Total FROM categoria;
```

categoria (1r × 1c)	
Total	
6	

### 9. Mostrar la cantidad de productos agrupados por categoría

```
SELECT Id_categoria, COUNT(SKU) AS Total FROM producto GROUP BY Id_categoria;
```

producto (4r × 2c)		
Id_categoria		Total
1		3
2		2
3		1
4		3

### 10. Mostrar el precio máximo y precio mínimo de los productos agrupados por categoría

```
SELECT ID_categoria, MAX(Precio) AS Precio_Maximo, MIN(Precio) AS Precio_Minimo  
FROM producto  
GROUP BY ID_categoria;
```

producto (4r × 3c)			
ID_categoria		Precio_Maximo	Precio_Minimo
1		3,100	1,400
2		39,000	12,400
3		150	150
4		16,000	4,100

### 11. Mostrar el promedio de los precios de los productos agrupados por categoría

```
SELECT ID_categoria, AVG(precio) AS Promedio_precios FROM producto GROUP BY  
ID_categoria;
```

producto (4r × 2c)	
ID_categoria	Promedio_precios
1	2,000
2	25,700
3	150
4	8,700

12. Mostrar la cantidad de productos agrupados por proveedor

```
SELECT ID_proveedor, COUNT(SKU) AS Total FROM producto GROUP BY ID_proveedor;
```

producto (3r × 2c)	
ID_proveedor	Total
00342	4
32910	3
39471	2

13. Mostrar la suma de los precios de los productos cuya categoría sea igual a 3

```
SELECT SUM(Precio) AS Suma_total FROM producto WHERE ID_categoria = '3';
```

producto (1r × 1c)	
Suma_total	
150	

14. Mostrar los productos que no pertenecen a la categoría 1 y 2

```
SELECT Nombre, Precio FROM producto WHERE Id_categoria NOT IN ('1', '2');
```

producto (4r × 2c)	
Nombre	Precio
Iphone 12	16,000
Xaomi Redmi Note 9	4,100
Samsung Galaxy s7	6,000
USB Kingston	150

15. Que nos muestre los clientes agrupados por ciudad a excepción de aquellos que están en Querétaro



```
SELECT Ciudad, COUNT(ID) FROM cliente GROUP BY Ciudad HAVING Ciudad <> 'Queretaro';
```

cliente (2r × 2c)	
Ciudad	COUNT(ID)
Boca del rio	1
Oaxaca	1

16. Que nos muestre los clientes agrupados por comuna a excepción de aquellos que se ubican en fuentes villanueva

```
SELECT Comuna, COUNT(ID) AS Total FROM cliente GROUP BY Comuna HAVING Comuna <> 'Fuentes villanueva';
```

cliente (2r × 2c)	
Comuna	Total
Fuentes Balvanera	1
la playita	1

17. Que nos muestre el total de números telefónicos de los clientes agrupados por nombre a excepción del cliente grupoVentura

```
SELECT cliente.Nombre, count(telefono_cliente.telefono) AS Total_Telefonos FROM cliente, telefono_cliente WHERE cliente.ID = telefono_cliente.ID_cliente GROUP BY cliente.Nombre HAVING cliente.Nombre <> 'grupoVentura';
```

cliente (2r × 2c)	
Nombre	Total_Telefonos
Chedraui	3
Siemens	2

18. Mostrar la cantidad de productos que se encuentran en la categoría AUDIO

```
SELECT COUNT(producto.ID_categoria) AS Total_productos FROM producto, categoria WHERE producto.ID_categoria = categoria.ID AND categoria.nombre = 'Audio';
```

producto (1r × 1c)	
Total_productos	
3	

19. Mostrar el nombre, precio de los productos que abasteció el proveedor TecnoAUDIO y del proveedor que muestre el teléfono y pagina web

```
SELECT producto.Nombre, producto.Precio, proveedor.Telefono, proveedor.Pagina_web
FROM producto, proveedor
WHERE producto.ID_proveedor = proveedor.ID AND proveedor.Nombre = 'TecnoAUDIO';
```

Resultado #1 (3r x 4c)			
Nombre	Precio	Telefono	Pagina_web
JBL	3,100	2881236745	www.TecnoEUDIO.com
Beat	1,500	2881236745	www.TecnoEUDIO.com
Audifonos Bose	1,400	2881236745	www.TecnoEUDIO.com

20. Mostrar el nombre de los productos que se le vendieron al cliente siemens y la cantidad que se llevó de cada uno

```
SELECT producto.Nombre, detalle_venta.Cantidad FROM producto, cliente, venta,
detalle_venta
WHERE detalle_venta.SKU = producto.SKU AND detalle_venta.Num_factura =
venta.Numero_factura
AND venta.ID_cliente = cliente.ID AND cliente.Nombre = 'Siemens';
```

Resultado #1 (1r x 2c)	
Nombre	Cantidad
Samsung Galaxy s7	5

21. Mostrar el nombre del cliente que se le hizo una venta en la fecha 2022-11-05 a las 6: 50 horas y mostrar cual fue el monto final

```
SELECT cliente.Nombre, venta.Monto_final FROM cliente, venta
WHERE venta.ID_cliente = cliente.ID AND venta.Fecha = '2022-11-05 6:50';
```

Resultado #1 (1r x 2c)	
Nombre	Monto_final
Siemens	15,700

22. Mostrar el teléfono de los clientes que se le hizo una venta en la fecha 2022-10-28 a las 12: 50 horas y que su monto fue de 23,150

```
SELECT telefono_cliente.Telefono FROM venta, cliente, telefono_cliente
WHERE cliente.ID = telefono_cliente.ID_cliente AND venta.ID_cliente = cliente.ID
AND venta.Fecha = '2022-10-28 12:50:00'
AND venta.Monto_final = 23150;
```

telefono_cliente (3r × 1c)	
Telefono	
2881394040	
4425940603	
4423742129	

23. Mostrar el nombre y precio de los productos que se agregaron al número de factura numero 5

```
SELECT producto.nombre, detalle_venta.Precio FROM detalle_venta, producto
WHERE detalle_venta.SKU = producto.SKU AND detalle_venta.Num_factura = 5;
```

Resultado #1 (3r × 2c)	
nombre	Precio
Samsung Galaxy s7	6,100
Asus	6,100
JBL	3,200

24. Mostrar el monto total de los productos que se agregaron al número de factura 5

```
SELECT SUM(detalle_venta.Importe) FROM detalle_venta, producto WHERE
detalle_venta.SKU = producto.SKU
AND detalle_venta.Num_factura = 5;
```

detalle_venta (1r × 1c)	
Monto_total	
56,400	

## SUBCONSULTAS

25. Mostrar los nombres de los productos que no pertenecen a la categoría de audio

```
SELECT producto.nombre FROM producto WHERE producto.ID_categoria
NOT IN (SELECT ID FROM categoria WHERE Nombre = 'AUDIO');
```

producto (5r × 2c)	
SKU	Nombre
1DFFG	Iphone 12
2VBNM	Xaomi Redmi Note 9
3RT9V	Beat
67CVN	USB Kingston
J45DR	Audifonos Bose

26. Mostrar los datos de los productos que pertenecen a la categoría de almacenamiento

```
SELECT * FROM producto WHERE producto.ID_categoria
IN (SELECT ID FROM categoria WHERE Nombre = 'ALMACENAMIENTO');
```

producto (1r × 6c)						
SKU	Nombre	Stock	Precio	ID_categoria	ID_proveedor	
67CVN	USB Kingston	70	150	3	00342	

27. Mostrar el código y el nombre del producto que no han sido vendidos

```
SELECT SKU, Nombre FROM producto WHERE producto.SKU
IN (SELECT detalle_venta.SKU FROM detalle_venta);
```

producto (4r × 2c)	
SKU	Nombre
09FGV	LENOVO
2CS13	Asus
2R46G	JBL
4HV0N	Samsung Galaxy s7

28. Mostrar todos los datos de los productos que no han sido vendidos

```
SELECT * FROM producto WHERE producto.SKU
NOT IN (SELECT detalle_venta.SKU FROM detalle_venta);
```

producto (5r × 6c)					
SKU	Nombre	Stock	Precio	ID_categoria	ID_proveedor
1DFFG	Iphone 12	10	16,000	4	39471
2VBNM	Xaomi Redmi Note 9	5	4,100	4	00342
3RT9V	Beat	100	1,500	1	32910
67CVN	USB Kingston	70	150	3	00342
J45DR	Audifonos Bose	30	1,400	1	32910

29. Mostrar la cantidad de productos que se adjuntaron al numero de factura 5 agrupados por nombre

```
SELECT producto.Nombre, SUM(detalle_venta.Cantidad) As Total FROM producto,
detalle_venta
WHERE producto.SKU = detalle_venta.SKU
AND producto.SKU IN (SELECT detalle_venta.SKU FROM detalle_venta WHERE
detalle_venta.Num_factura = 5) GROUP BY producto.Nombre;
```

producto (3r × 2c)	
Nombre	Total
Asus	3
JBL	10
Samsung Galaxy s7	5

30. Mostrar el precio de los productos que se le vendieron al cliente con el ID = 43579

```
SELECT detalle_venta.Precio FROM detalle_venta WHERE detalle_venta.Num_factura
IN (SELECT venta.Numero_factura FROM venta WHERE venta.ID_cliente = '43579');
```

detalle_venta (3r × 1c)	
Precio	
6,100	
6,100	
3,200	

## PROCEDIMIENTOS ALMACENADOS

- Procedimientos almacenados categoría

```
CREATE PROCEDURE AddCategoria(_nombre VARCHAR(100), _descripcion TEXT)
BEGIN
    DECLARE Estatus VARCHAR(20);
    SET Estatus = 'Activo';
    INSERT INTO categoria VALUES (NULL, _nombre, _descripcion, Estatus);
END$$
```

```
CREATE PROCEDURE DeleteCategoria(_id INT, _estatus ENUM('Activo', 'Inactivo'))
BEGIN
    UPDATE categoria SET Estatus = _estatus WHERE ID = _id;
END$$
```

```
CREATE PROCEDURE UpdateCategoria(_Id INT, _nombre VARCHAR(100), _descripcion TEXT)
BEGIN
    UPDATE categoria SET Nombre = _nombre, Descripcion = _descripcion WHERE ID = _Id;
END$$
```

- Procedimientos almacenados cliente

```
CREATE PROCEDURE AddCliente
(_id CHAR(5), _nombre VARCHAR(80), _email VARCHAR(100), _cp CHAR(5), _ciudad
VARCHAR(100), _comuna VARCHAR(100), _calle VARCHAR(100), _numero INT)
BEGIN
    DECLARE Estatus VARCHAR(20);
    SET Estatus = 'Activo';
    INSERT INTO cliente VALUES (_id, _nombre, _email, _cp, _ciudad, _comuna, _calle,
_numero, _estatus);
END$$
```

```
CREATE PROCEDURE DeleteCliente
(_id CHAR(5), _estatus ENUM('Activo', 'Inactivo'))
BEGIN
    UPDATE cliente SET Estatus = _estatus WHERE ID = _id;
END$$
```

```
CREATE PROCEDURE UpdateCliente
(_id CHAR(5), _nombre VARCHAR(100), _email VARCHAR(100), _cp CHAR(5), _ciudad
VARCHAR(100), _comuna VARCHAR(100), _calle VARCHAR(100), _numero INT)
BEGIN
    UPDATE cliente
    SET Nombre = _nombre, Email = _email, CP = _cp,
    Ciudad = _ciudad, Comuna = _comuna, Calle = _calle, Numero = _numero
    WHERE ID = _id;
END$$
```

- Procedimientos almacenados productos

```
CREATE PROCEDURE AddProducto
(_sku VARCHAR(5), _nombre VARCHAR(100), _stock INT, _precio DOUBLE, _precio_venta
DOUBLE, _id_categoria INT, _id_proveedor CHAR(5))
BEGIN
    DECLARE Estatus VARCHAR(20);
    SET Estatus = 'Activo';
    INSERT INTO producto VALUES (_sku, _nombre, _stock, _precio, _precio_venta,
_id_categoria, _id_proveedor, Estatus);
END$$
```

```
CREATE PROCEDURE DeleteProducto (_sku VARCHAR(5), _estatus ENUM('Activo',
'Inactivo'))
BEGIN
    UPDATE producto SET Estatus = _estatus WHERE SKU = _sku;
END$$
```

```
CREATE PROCEDURE UpdateProducto
(_sku VARCHAR(5), _nombre VARCHAR(100), _stock INT, _precio DOUBLE, _id_categoria
INT, _id_proveedor CHAR(5))
BEGIN
    UPDATE producto SET Nombre = _nombre, Stock = _stock, Precio = _precio,
ID_categoria = _id_categoria, ID_proveedor = _id_proveedor WHERE SKU = _sku;
END$$
```

- Procedimientos almacenados proveedor

```
CREATE PROCEDURE AddProveedor(_id CHAR(5), _nombre VARCHAR(100), _rfc VARCHAR(10),
_telefono CHAR(10), _direccion VARCHAR(100), _pagina_web TEXT)
BEGIN
    DECLARE Estatus VARCHAR(20);
    SET Estatus = 'Activo';
    INSERT INTO proveedor VALUES(_id, _nombre, _rfc, _telefono, _direccion,
_pagina_web, Estatus);
END$$
```

```
CREATE PROCEDURE DeleteProveedor (_id CHAR(5), _estatus ENUM('Activo', 'Inactivo'))
BEGIN
    UPDATE proveedor SET Estatus = _estatus WHERE ID = _id;
END$$
```

```
CREATE PROCEDURE UpdateProveedor (_id CHAR(5), _nombre VARCHAR(100), _rfc
VARCHAR(10), _telefono CHAR(10), _direccion VARCHAR(100), _pagina_web TEXT)
BEGIN
    UPDATE proveedor SET Nombre = _nombre, RFC = _rfc, Telefono = _telefono,
Direccion = _direccion, Pagina_web = _pagina_web WHERE ID = _id;
END$$
```

- Procedimientos almacenados Telefono\_cliente

```
CREATE PROCEDURE AddTelefono_cliente (_id_cliente CHAR(5), _telefono CHAR(10))
BEGIN
```

```
INSERT INTO telefono_cliente VALUES(_id_cliente, _telefono);
END$$
```

```
CREATE PROCEDURE DeleteTelefono_cliente (_id_cliente CHAR(5) ,_telefono CHAR(10))
BEGIN
    DELETE FROM telefono_cliente WHERE ID_cliente = _id_cliente AND Telefono =
    _telefono;
END$$
```

- Procedimientos almacenados detalle\_venta

```
CREATE PROCEDURE AddDetalle_venta (_num_factura INT, _sku_producto VARCHAR(10),
    _cantidad INT, _precio DOUBLE)
BEGIN
    DECLARE importe DOUBLE;
    SELECT CalcularImporte(_cantidad, _precio) INTO importe;
    INSERT INTO detalle_venta VALUES(_num_factura, _sku_producto, _cantidad, _precio,
    importe);
END$$
```

- Procedimientos almacenados de venta

```
CREATE PROCEDURE AddVenta(_fecha DATETIME, _id_cliente CHAR(5), _monto_final
    DOUBLE)
BEGIN
    INSERT INTO venta VALUES(_fecha, _id_cliente, _monto_final);
END$$
```

- Procedimientos almacenados usuarios

```
CREATE PROCEDURE AddUser(_id CHAR(5), _usuario VARCHAR(50), _rol
    ENUM('SuperAdmin','Administrador', 'Vendedor'), _contraseña VARCHAR(30))
BEGIN
    DECLARE estatus VARCHAR(20);
    SET estatus = 'Activo';
    INSERT INTO users VALUES (_id, _usuario, _rol, _contraseña, estatus);
END$$
```

```
CREATE PROCEDURE DeleteUser(_id CHAR(5), _estatus ENUM('Activo', 'Inactivo'))
BEGIN
    UPDATE users SET Estatus = _estatus WHERE ID = _id;
END$$
```

```
CREATE PROCEDURE UpdateUser(_id CHAR(5), _usuario VARCHAR(50), _rol
    ENUM('SuperAdmin','Administrador', 'Vendedor'), _contraseña VARCHAR(30))
BEGIN
    UPDATE users SET Usuario = _usuario, Rol_usuario = _rol, Contraseña = _contraseña
    WHERE ID = _id;
END$$
```



- Procedimientos almacenados para cargar los datos en un datagridview de las interfaces

```
CREATE PROCEDURE dgvCategorias()  
BEGIN  
    SELECT * FROM categoria;  
END$$  
  
CREATE PROCEDURE dgvProveedores()  
BEGIN  
    SELECT * FROM proveedor;  
END  
  
CREATE PROCEDURE dgvCliente()  
BEGIN  
    SELECT * FROM cliente;  
END$$  
  
CREATE PROCEDURE dgvProductos()  
BEGIN  
    SELECT * FROM producto;  
END$$  
  
CREATE PROCEDURE dgvTelefono_cliente()  
BEGIN  
    SELECT telefono_cliente.ID_cliente, cliente.Nombre, telefono_cliente.Telefono  
    FROM telefono_cliente, cliente  
    WHERE cliente.ID = telefono_cliente.ID_cliente;  
END$$  
  
CREATE PROCEDURE dgvUsuarios()  
BEGIN  
    SELECT * FROM users;  
END$$  
  
CREATE PROCEDURE dgvBitacora()  
BEGIN  
    SELECT * FROM bitacora;  
END$$  
  
CREATE PROCEDURE dgvCompras()  
BEGIN  
    SELECT * FROM detalle_venta;  
END$$  
  
CREATE PROCEDURE dgvVentas()  
BEGIN  
    SELECT * FROM venta;  
END$$
```

```
CREATE PROCEDURE dgvCortes()  
BEGIN  
    SELECT * FROM corte_caja;  
END$$
```

## FUNCIONES

Esta función la utilizamos para calcular el importe por cada venta de producto dependiendo la cantidad y precio

```
CREATE FUNCTION CalcularImporte(_cantidad INT, _precio DOUBLE)  
RETURNS DOUBLE  
BEGIN  
    RETURN _cantidad * _precio;  
END$$
```

## DISPARADORES

El siguiente disparador se creó para restar el stock dependiendo la cantidad de venta

```
CREATE TRIGGER update_stock AFTER INSERT ON detalle_venta FOR EACH ROW  
BEGIN  
    DECLARE stock_temp INT;  
    SET stock_temp = (SELECT Stock FROM producto WHERE SKU = NEW.SK) - NEW.Cantidad;  
    if (stock_temp >= 0) then  
        UPDATE producto SET Stock = stock_temp WHERE SKU = NEW.SKU;  
    END if;  
END$$
```

## USUARIOS

El sistema contara con los siguientes usuarios:

- 🔑 Vendedor: Solo tendrá acceso a las ventas, consulta de productos y clientes.
- 🔑 Administrador: El administrador tendrá acceso a todo menos a el control de usuarios
- 🔑 SuperAdmin: Naturalmente el superAdmin tendrá acceso a todos los datos del sistema.