****

**ISW-312 Fundamentos de Bases de Datos**

Presentado por:

**Sebastian Alonso Rodriguez Varela**

Profesor

**Freddy Gerardo Rocha Boza**

Costa Rica, Ciudad Quesada

2024

# Tabla de Contenido

Contenido

[Tabla de Contenido 2](#_Toc164798249)

[Introducción 3](#_Toc164798250)

[Planteamiento del problema 4](#_Toc164798251)

[Justificación 5](#_Toc164798252)

[Objetivos 6](#_Toc164798253)

[Objetivo general: 6](#_Toc164798254)

[Objetivos específicos: 6](#_Toc164798255)

[Descripción de la Solución 7](#_Toc164798256)

[Usuarios: 7](#_Toc164798257)

[Proveedores: 7](#_Toc164798258)

[Clientes: 7](#_Toc164798259)

[Direcciones Clientes: 7](#_Toc164798260)

[Direcciones Proveedores: 7](#_Toc164798261)

[Contactos Clientes: 8](#_Toc164798262)

[Contactos Proveedores: 8](#_Toc164798263)

[Contactos Usuarios: 8](#_Toc164798264)

[Facturas: 8](#_Toc164798265)

[Ventas: 8](#_Toc164798266)

[Productos: 8](#_Toc164798267)

[Detalles Venta: 8](#_Toc164798268)

[Categorías Productos: 9](#_Toc164798269)

[Producto Categoría: 9](#_Toc164798270)

[Diagrama E/R 10](#_Toc164798271)

[Scripts 11](#_Toc164798272)

[Tablas de la base de datos: 11](#_Toc164798273)

[Inserción de datos 13](#_Toc164798274)

[Consultas básicas 15](#_Toc164798275)

[8. Realizar un total de 10 script de eliminación y actualización de datos 16](#_Toc164798276)

[9. y 10. Consultas para repórtes 17](#_Toc164798277)

[Conclusión 19](#_Toc164798278)

[**Bibliografia** 20](#_Toc164798279)

# Introducción

Independientemente de su tamaño, cualquier empresa necesita gestionar bien su inventario y automatizar sus procesos para tener éxito. La microempresa de Accesorios para Autos RV de Ciudad Quesada encuentra importantes dificultades para gestionar su stock y mejorar sus flujos de trabajo.

RV Car accesorios ofrece una amplia gama de accesorios para automóviles, con especial atención a tapacubos de varias marcas. A pesar de los desafíos de la empresa para mantener su inventario, tiene dificultades para administrar el stock y cumplir con los pedidos de manera efectiva. En la actualidad, el inventario se maneja en papel.

El objetivo de este proyecto es crear y utilizar una base de datos que ayude a RV Car accesorios a realizar un seguimiento de sus productos, restringir quién puede ver los datos de forma segura y realizar facturas.

Este documento describe un plan integral para establecer la base de datos de accesorios para automóviles RV, cuyo objetivo es resolver los problemas existentes de gestión de inventario y mejorar la eficacia operativa general de la empresa.

# Enunciado del problema

RV Car Accesorios es un pequeño negocio ubicado en Ciudad Quesada que se especializa en la venta de accesorios para automóviles. La empresa ofrece una amplia gama de productos, incluidos modelos y variantes únicos en términos de tamaño, color, diseño y año de producción. Su producto más son los tapacubos de varias marcas. Sin embargo, tiene problemas para mantener sus existencias al día, lo que constituye un obstáculo importante, ya que actualmente se maneja un inventario en papel y memoria del dueño.

La problemática principal radica en la gestión del inventario. Debido a la gran cantidad de variantes de productos y su constante cambio, mantener un registro actualizado se convierte en una tarea compleja. Además, la empresa necesita implementar características que le permitan funcionar eficientemente en aspectos de sistemas de red, control de accesos y generación de facturas.

# Justificación

La creación de una base de datos adecuada es crucial para abordar los desafíos de gestión de inventario y mejorar la eficiencia operativa de " RV Car Accesorios ". La implementación de un sistema de gestión de bases de datos permitirá:

1. Gestión de Inventarios Eficiente: Una base de datos centralizada facilitará el seguimiento y actualización de inventarios, lo que ayudará a evitar errores y pérdidas.
2. Control de Acceso y Seguridad: La base de datos permitirá establecer perfiles de usuarios y controlar el acceso a la información de manera segura, garantizando la confidencialidad y la integridad de los datos.
3. Generación de Facturas Automatizada: Al integrar funciones de facturación en la base de datos, se simplificará el proceso de generación de facturas, agilizando las transacciones comerciales.
4. Análisis de Datos y Tendencias: La base de datos proporcionará información valiosa sobre las ventas, preferencias de los clientes y tendencias del mercado, lo que facilitará la toma de decisiones estratégicas.

# Objetivos

## Objetivo general:

Se a de crear una base de datos que permita el funcionamiento de la microempresa, para esto es necesario el manejar bien los datos

## Objetivos específicos:

Crear una base de datos con datos necesarios para la gestión de inventarios

Permitir el funcionar a la microempresa en cuanto a inventarios

Dar las funcionalidades de crear facturas

# Descripción de la Solución

Para resolver el problema planteado, se crearon varias tablas que almacenan la información de manera organizada y eficiente con el objetivo de cumplir los puntos planteados de acuerdo con las necesidades del cliente y utilizando las normas de bases de datos.

A continuación, se describe cada tabla y su función:

## Usuarios:

1. Almacena la información de los usuarios del sistema, como cédula, nombre, apellidos, teléfono y correo electrónico.
2. La cédula se utiliza como clave principal para identificar a cada usuario de manera única.

## Proveedores:

1. Registra los proveedores de productos, incluyendo su ID, nombre, producto que proveen, dirección, teléfono, correo electrónico y descripción.
2. Cada proveedor puede tener múltiples direcciones y contactos asociados.

## Clientes:

1. Almacena los datos de los clientes que realizan compras, incluyendo su cédula, nombre, teléfono, empresa (si aplica) y correo electrónico.
2. La cédula se utiliza como clave principal para identificar a cada cliente de manera única.

### Direcciones Clientes:

1. Guarda las direcciones asociadas a los clientes, con un ID único, la cédula del cliente y la dirección.
2. Permite tener múltiples direcciones asociadas a un mismo cliente.

### Direcciones Proveedores:

1. Almacena las direcciones de los proveedores, con un ID único, el ID del proveedor y la dirección.
2. Permite tener múltiples direcciones asociadas a un mismo proveedor.

### Contactos Clientes:

1. Contiene los detalles de contacto de los clientes, como ID único, cédula del cliente, teléfono y correo electrónico.
2. Permite tener múltiples contactos asociados a un mismo cliente.

## Contactos Proveedores:

1. Registra los contactos de los proveedores, incluyendo su ID único, ID del proveedor, teléfono y correo electrónico.
2. Permite tener múltiples contactos asociados a un mismo proveedor.

## Contactos Usuarios:

1. Almacena los detalles de contacto de los usuarios, incluyendo su ID único, cédula de usuario, teléfono y correo electrónico.
2. Permite tener múltiples contactos asociados a un mismo usuario.

## Facturas:

1. Registra las facturas generadas por las ventas, con un número único de venta, fecha, monto, cédula del cliente y cédula del usuario.
2. Permite relacionar las facturas con los clientes y usuarios correspondientes.

## Ventas:

1. Guarda los registros de las ventas realizadas, con un número único de venta, fecha, monto, medio de pago, cédula del cliente y cédula del usuario.
2. Permite relacionar las ventas con los clientes y usuarios correspondientes.

## Productos:

1. Almacena la información de los productos disponibles para la venta, incluyendo ID único, marca, tamaño, color, peso, precio, estilo y descripción.
2. Permite identificar de manera única cada producto.

## Detalles Venta:

1. Guarda los detalles de las ventas, normalizando la relación entre ventas y productos, con un ID único, número de venta, ID del producto vendido, cantidad y precio.
2. Permite relacionar los detalles de venta con las ventas y productos correspondientes.

## Categorías Productos:

1. Registra las categorías a las que pertenecen los productos, con un ID único, nombre y descripción.
2. Permite clasificar los productos en distintas categorías.

## Producto Categoría:

1. Establece la relación entre productos y categorías, con el ID del producto y el ID de la categoría.
2. Permite asociar múltiples categorías a un mismo producto y viceversa.

# Diagrama E/R

En dicha imagen se puede ver el diagrama de las tablas de la base de datos.

Una captura de pantalla de una red social

Descripción generada automáticamente

# Scripts

## Tablas de la base de datos:

Estas fueron descritas previamente en este documento

--Tabla de Usuarios utilizada para almacenar a los empleados y usuarios del sistema

CREATE TABLE Usuarios (

cedula VARCHAR(20) CONSTRAINT PK\_Cedula\_Usuarios PRIMARY KEY,

nombre VARCHAR(50),

apellidos VARCHAR(50),

telefono VARCHAR(20),

correo VARCHAR(100)

);

--Tabla de Proveedores, esta permite el almacenar los proveedores

CREATE TABLE Proveedores (

id INT IDENTITY(1,1) CONSTRAINT PK\_ID\_Proveedores PRIMARY KEY,

nombre VARCHAR(50),

nom\_producto VARCHAR(50),

direccion VARCHAR(100),

telefono VARCHAR(20),

correo VARCHAR(100),

descripcion VARCHAR(80)

);

--Esta tabla Clientes permite el almacenar la informacion de los clientes

CREATE TABLE Clientes (

cedula VARCHAR(20) CONSTRAINT PK\_Cedula\_Clientes PRIMARY KEY,

nombre VARCHAR(50),

telefono VARCHAR(20),

empresa VARCHAR(50),

correo VARCHAR(100)

);

--Esta tabla permite el almacenar los datos de las direcciones de los clientes

CREATE TABLE DireccionesClientes (

id\_direccion INT IDENTITY(1,1) CONSTRAINT PK\_ID\_DireccionesClientes PRIMARY KEY,

cedula\_cliente VARCHAR(20),

direccion VARCHAR(80),

CONSTRAINT FK\_Cedula\_Clientes\_DireccionesClientes FOREIGN KEY (cedula\_cliente) REFERENCES Clientes(cedula)

);

--Esta tabla permite el almacenar la información de las direcciones de los proveedores

CREATE TABLE DireccionesProveedores (

id\_direccion INT IDENTITY(1,1) CONSTRAINT PK\_ID\_DireccionesProveedores PRIMARY KEY,

id\_proveedor INT,

direccion VARCHAR(80),

CONSTRAINT FK\_ID\_Proveedores\_DireccionesProveedores FOREIGN KEY (id\_proveedor) REFERENCES Proveedores(id)

);

--Esta tabla almacena los contactos de los clientes

CREATE TABLE ContactosClientes (

id\_contacto INT IDENTITY(1,1) CONSTRAINT PK\_ID\_ContactosClientes PRIMARY KEY,

cedula\_cliente VARCHAR(20),

telefono VARCHAR(20),

correo VARCHAR(100),

CONSTRAINT FK\_Cedula\_Clientes\_ContactosClientes FOREIGN KEY (cedula\_cliente) REFERENCES Clientes(cedula)

);

--Esta tabla Contactos Proveedores permite el almacenar los contactos e los proveedores

CREATE TABLE ContactosProveedores (

id\_contacto INT IDENTITY(1,1) CONSTRAINT PK\_ID\_ContactosProveedores PRIMARY KEY,

id\_proveedor INT,

telefono VARCHAR(20),

correo VARCHAR(100),

CONSTRAINT FK\_ID\_Proveedores\_ContactosProveedores FOREIGN KEY (id\_proveedor) REFERENCES Proveedores(id)

);

--Esta tabla contactos usuarios permite el almacenar los contactos de los usuarios u empleados

CREATE TABLE ContactosUsuarios (

id\_contacto INT IDENTITY(1,1) CONSTRAINT PK\_ID\_ContactosUsuarios PRIMARY KEY,

cedula\_usuario VARCHAR(20),

telefono VARCHAR(20),

correo VARCHAR(100),

CONSTRAINT FK\_Cedula\_Usuarios\_ContactosUsuarios FOREIGN KEY (cedula\_usuario) REFERENCES Usuarios(cedula)

);

--Esta tabla permite almacenar la información de las facturas

-- Facturas

CREATE TABLE Facturas (

numero\_venta INT CONSTRAINT PK\_Numero\_Venta\_Facturas PRIMARY KEY,

fecha DATETIME,

monto DECIMAL(10, 2),

cedula\_cliente VARCHAR(20),

cedula\_usuario VARCHAR(20),

CONSTRAINT FK\_Cedula\_Clientes\_Facturas FOREIGN KEY (cedula\_cliente) REFERENCES Clientes(cedula),

CONSTRAINT FK\_Cedula\_Usuarios\_Facturas FOREIGN KEY (cedula\_usuario) REFERENCES Usuarios(cedula)

);

--Esta tabla Ventas permite almacenar la informacion de cada venta realizada, su función es ser un encabezado

CREATE TABLE Ventas (

numero\_venta INT CONSTRAINT PK\_Numero\_Venta\_Ventas PRIMARY KEY,

fecha DATETIME,

monto DECIMAL(10, 2),

medio\_pago VARCHAR(50),

cedula\_cliente VARCHAR(20),

cedula\_usuario VARCHAR(20),

CONSTRAINT FK\_Cedula\_Clientes\_Ventas FOREIGN KEY (cedula\_cliente) REFERENCES Clientes(cedula),

CONSTRAINT FK\_Cedula\_Usuarios\_Ventas FOREIGN KEY (cedula\_usuario) REFERENCES Usuarios(cedula)

);

-- Productos

CREATE TABLE Productos (

id\_producto INT IDENTITY(1,1) CONSTRAINT PK\_ID\_Productos PRIMARY KEY,

marca VARCHAR(50),

tamanio VARCHAR(50),

color VARCHAR(50),

peso DECIMAL(5,2),

precio DECIMAL(10,2),

estilo VARCHAR(50),

descripcion VARCHAR(80)

);

--Esta tabla Detalle Ventas almacena la información sobre el producto vendido y monto

-- DetallesVenta (Nueva tabla para normalizar Ventas)

CREATE TABLE DetallesVenta (

id\_detalle INT IDENTITY(1,1) CONSTRAINT PK\_ID\_DetallesVenta PRIMARY KEY,

numero\_venta INT,

id\_producto INT,

cantidad INT,

precio DECIMAL(10, 2),

CONSTRAINT FK\_Numero\_Venta\_DetallesVenta FOREIGN KEY (numero\_venta) REFERENCES Ventas(numero\_venta),

CONSTRAINT FK\_ID\_Productos\_DetallesVenta FOREIGN KEY (id\_producto) REFERENCES Productos(id\_producto)

);

--Esta tabla Categorías Productos permite el almacenar la información referente a las categorías de los productos

-- CategoriasProductos

CREATE TABLE CategoriasProductos (

id\_categoria INT IDENTITY(1,1) CONSTRAINT PK\_ID\_CategoriasProductos PRIMARY KEY,

nombre VARCHAR(100),

descripcion VARCHAR(80)

);

--Esta tabla Producto Categoria permite el crear una relación en el sistema si existe mas de una categoría para algun producto y facilitar dicha tarea y evitar repetir informacion

-- Relación entre productos y categorías (si un producto puede tener varias categorías)

CREATE TABLE ProductoCategoria (

id\_producto INT,

id\_categoria INT,

CONSTRAINT PK\_ID\_ProductoCategoria PRIMARY KEY (id\_producto, id\_categoria),

CONSTRAINT FK\_ID\_Productos\_ProductoCategoria FOREIGN KEY (id\_producto) REFERENCES Productos(id\_producto),

CONSTRAINT FK\_ID\_CategoriasProductos\_ProductoCategoria FOREIGN KEY (id\_categoria) REFERENCES CategoriasProductos(id\_categoria)

);

## Inserción de datos

Estos scrips tienen la finalidad de demostrar el funcionamiento de las tablas y su correcta creacion

-- Inserción en Usuarios

INSERT INTO Usuarios (cedula, nombre, apellidos, telefono, correo) VALUES

('101010101', 'Ana', 'Martínez', '1234-5678', 'ana.martinez@example.com'),

('202020202', 'Luis', 'González', '8765-4321', 'luis.gonzalez@example.com');

-- Inserción en Clientes

INSERT INTO Clientes (cedula, nombre, telefono, empresa, correo) VALUES

('303030303', 'Don Hose', '2345-6789', 'AutosDeSegunda', 'contacto@autosdesegunda.com'),

('404040404', 'Corporacion De Ganaderos S.A.', '9876-5432', 'Corporacion De Ganaderos S.A.', 'contacto@Corporacion De Ganaderos S.A.com');

-- Inserción en Proveedores

INSERT INTO Proveedores (nombre, nom\_producto, direccion, telefono, correo, descripcion) VALUES

('Accesorios Rojas', 'Tapacubos Toyota rojo 67mm', 'Calle la Ezperanza Esprza', '2345-6789', 'contacto@proveedoraOriente.com', 'Distribudor de aacesorios Toyota'),

('Importaciones del Oriente', 'Tapacubos Nissa plata de 63mm', 'Avenida no Vienvenida', '9876-5432', 'contacto@importacionesoriente12.com', 'Importadora de accesorios directos de china');

-- Inserción en ContactosClientes

INSERT INTO ContactosClientes (cedula\_cliente, telefono, correo) VALUES

('303030303', '2345-6789', 'contacto@autosdesegunda.com'),

('404040404', '9876-5432', 'contacto@Corporacion De Ganaderos S.A.com');

-- Inserción en ContactosProveedores

INSERT INTO ContactosProveedores (id\_proveedor, telefono, correo) VALUES

(1, '2345-6789', 'contacto@proveedoraOriente.com'),

(2, '9876-5432', 'contacto@importacionesoriente12.com');

-- Inserción en ContactosUsuarios

INSERT INTO ContactosUsuarios (cedula\_usuario, telefono, correo) VALUES

('101010101', '1234-5678', 'ana.martinez@example.com'),

('202020202', '8765-4321', 'luis.gonzalez@example.com');

-- Inserción en DireccionesClientes

INSERT INTO DireccionesClientes (cedula\_cliente, direccion) VALUES

('303030303', 'Calle 158, Cartago'),

('404040404', 'Avenida Principal 456, San Jose');

-- Inserción en DireccionesProveedores

INSERT INTO DireccionesProveedores (id\_proveedor, direccion) VALUES

(1, 'Avenida Siempre Viva 789, Ciudad C'),

(2, 'Calle Nueva 123, Ciudad D');

-- Inserción en Facturas

INSERT INTO Facturas (numero\_venta, fecha, monto, cedula\_cliente, cedula\_usuario) VALUES

(1, '2024-01-01 10:00:00', 500.00, '303030303', '101010101'),

(2, '2024-01-02 11:00:00', 750.00, '404040404', '202020202');

-- Inserción en Ventas

INSERT INTO Ventas (numero\_venta, fecha, monto, medio\_pago, cedula\_cliente, cedula\_usuario) VALUES

(1, '2024-01-01 10:00:00', 500.00, 'Efectivo', '303030303', '101010101'),

(2, '2024-01-02 11:00:00', 750.00, 'Tarjeta', '404040404', '202020202');

-- Inserción en Productos

INSERT INTO Productos (marca, tamanio, color, peso, precio, estilo, descripcion) VALUES

('Toyota', '67 mm', 'Rojo', 0.50, 100.00, 'Clásico', 'Tapacubos Estilo original color rojo de 67 mm 2000-2019'),

('Nissa', '63 mm', 'Plata', 0.67, 150.00, 'Moderno', 'Tapacubos Estilo B ,color plata de 63 mm 2007-2012');

-- Inserción en DetallesVenta

INSERT INTO DetallesVenta (numero\_venta, id\_producto, cantidad, precio) VALUES

(1, 1, 2, 100.00), -- Venta 1, Producto 1, 2 unidades

(2, 2, 3, 150.00); -- Venta 2, Producto 2, 3 unidades

-- Inserción en CategoriasProductos

INSERT INTO CategoriasProductos (nombre, descripcion) VALUES

('Tapacubos', 'Tapacubos para centro de rueda'),

('Toyota', 'Accesorio para autos marca Toyota'),

('NISSA', 'Accesorio para autos marca NIIISAN');

-- Inserción en Relación Producto-Categoría

INSERT INTO ProductoCategoria (id\_producto, id\_categoria) VALUES

(1, 1), -- Producto 1 en Categoría 1

(1, 2), -- Producto 1 en Categoría 2

(2, 1), -- Producto 2 en Categoría 1

(2, 2); -- Producto 2 en Categoría 2

## Consultas básicas

Con estas consultas se verifica el correcto funcionamiento de la tabla y afinamiento de las habilidades en bases de datos.

--- Seleccionar todos los datos de Usuarios

SELECT \* FROM Usuarios;

--- Seleccionar todos los datos de Clientes

SELECT \* FROM Clientes;

--- Seleccionar todos los datos de Facturas

SELECT \* FROM Facturas;

--- Seleccionar todos los datos de Ventas

SELECT \* FROM Ventas;

--- Seleccionar todos los datos de Productos

SELECT \* FROM Productos;

--- Seleccionar todos los datos de DetallesVenta

SELECT \* FROM DetallesVenta;

--- Seleccionar todos los datos de Proveedores

SELECT \* FROM Proveedores;

--- Seleccionar todos los datos de Contactos clientes

SELECT c.nombre, cc.telefono, cc.correo

FROM ContactosClientes cc

JOIN Clientes c ON cc.cedula\_cliente = c.cedula;

--- Seleccionar todos los datos de Contactos proveedores

SELECT p.nombre, cp.telefono, cp.correo

FROM ContactosProveedores cp

JOIN Proveedores p ON cp.id\_proveedor = p.id;

--- Seleccionar todos los datos de Contactos Usarios

SELECT u.nombre, u.apellidos, cu.telefono, cu.correo

FROM ContactosUsuarios cu

JOIN Usuarios u ON cu.cedula\_usuario = u.cedula;

--- Seleccionar todos los datos de Direcciones clientes

SELECT c.nombre, dc.direccion

FROM DireccionesClientes dc

JOIN Clientes c ON dc.cedula\_cliente = c.cedula;

--- Seleccionar todos los datos de Direcciones proveedores

SELECT p.nombre, dp.direccion

FROM DireccionesProveedores dp

JOIN Proveedores p ON dp.id\_proveedor = p.id;

--- Seleccionar todos los datos de CategoriasProductos

SELECT \* FROM CategoriasProductos;

--- Seleccionar todos los datos de la relación Producto-Categoría

SELECT \* FROM ProductoCategoria;

## 8. Realizar un total de 10 script de eliminación y actualización de datos

Estos son los scrips solicitas en el punto 8

------------------- Actualizar el nombre de un cliente

UPDATE Clientes

SET nombre = 'Nuevo Nombre'

WHERE cedula = '303030303';

------------------- Eliminar un cliente y sus direcciones asociadas

DELETE FROM DireccionesClientes

WHERE cedula\_cliente = '404040404';

DELETE FROM ContactosClientes

WHERE cedula\_cliente = '404040404';

DELETE FROM Clientes

WHERE cedula = '404040404';

------------------- Actualizar el teléfono de un proveedor:

UPDATE ContactosProveedores

SET telefono = '1111-1111'

WHERE id\_proveedor = 1;

UPDATE Proveedores

SET telefono = '1111-1111'

WHERE id = 1;

------------------- Eliminar un producto y sus detalles de venta asociados

DELETE FROM DetallesVenta

WHERE id\_producto = 2;

DELETE FROM Productos

WHERE id\_producto = 2;

------------------- Actualizar el correo electrónico de un usuario

UPDATE ContactosUsuarios

SET correo = 'nuevo\_correo@example.com'

WHERE cedula\_usuario = '101010101';

UPDATE Usuarios

SET correo = 'nuevo\_correo@example.com'

WHERE cedula = '101010101';

------------------- Eliminar un contacto de proveedor específico

DELETE FROM ContactosProveedores

WHERE id\_contacto = 1;

------------------- Actualizar la descripción de un proveedor:

UPDATE Proveedores

SET descripcion = 'Nueva descripción'

WHERE id = 1;

------------------- Eliminar una categoría de productos y sus relaciones asociadas:

DELETE FROM ProductoCategoria

WHERE id\_categoria = 1;

DELETE FROM CategoriasProductos

WHERE id\_categoria = 1;

------------------- Actualizar la dirección de un cliente:

UPDATE DireccionesClientes

SET direccion = 'Nueva Dirección'

WHERE id\_direccion = 1;

------------------- Eliminar una venta y sus detalles asociados:

DELETE FROM DetallesVenta

WHERE numero\_venta = 1;

DELETE FROM Ventas

WHERE numero\_venta = 1;

# 9. y 10. Consultas para repórtes

Aquí se realizan las consultas solicitas en el punto 9 y 10 del documento

---------------Cantidad de productos vendidos por cliente:

SELECT c.nombre AS Nombre\_Cliente, COUNT(dv.id\_detalle) AS Cantidad\_Productos\_Vendidos

FROM Clientes c

JOIN Ventas v ON c.cedula = v.cedula\_cliente

JOIN DetallesVenta dv ON v.numero\_venta = dv.numero\_venta

GROUP BY c.nombre;

------------------Total de ventas mensuales por usuario:

SELECT MONTH(v.fecha) AS Mes, YEAR(v.fecha) AS Año, u.nombre AS Nombre\_Usuario, SUM(v.monto) AS Total\_Ventas

FROM Ventas v

JOIN Usuarios u ON v.cedula\_usuario = u.cedula

GROUP BY MONTH(v.fecha), YEAR(v.fecha), u.nombre;

------------------Promedio de precio de los productos por categoría:

SELECT cp.nombre AS Categoria, AVG(p.precio) AS Promedio\_Precio

FROM CategoriasProductos cp

JOIN ProductoCategoria pc ON cp.id\_categoria = pc.id\_categoria

JOIN Productos p ON pc.id\_producto = p.id\_producto

GROUP BY cp.nombre;

------------------Total de ventas por marca de producto:

SELECT p.marca AS Marca, SUM(dv.cantidad) AS Total\_Vendido

FROM Productos p

JOIN DetallesVenta dv ON p.id\_producto = dv.id\_producto

GROUP BY p.marca;

--------------------Total de ventas realizadas en efectivo por cliente:

SELECT c.nombre AS Nombre\_Cliente, SUM(v.monto) AS Total\_Ventas\_Efectivo

FROM Clientes c

JOIN Ventas v ON c.cedula = v.cedula\_cliente

WHERE v.medio\_pago = 'Efectivo'

GROUP BY c.nombre;

------------------Precio máximo de los productos por marca:

SELECT p.marca AS Marca, MAX(p.precio) AS Precio\_Maximo

FROM Productos p

GROUP BY p.marca;

---------------Cantidad de productos vendidos por categoría:

SELECT cp.nombre AS Categoria, COUNT(dv.id\_detalle) AS Cantidad\_Productos\_Vendidos

FROM CategoriasProductos cp

JOIN ProductoCategoria pc ON cp.id\_categoria = pc.id\_categoria

JOIN Productos p ON pc.id\_producto = p.id\_producto

JOIN DetallesVenta dv ON p.id\_producto = dv.id\_producto

GROUP BY cp.nombre;

---------------------Total de ventas por cliente y medio de pago

SELECT c.nombre AS Cliente, v.medio\_pago AS Medio\_de\_Pago, SUM(v.monto) AS Total\_Ventas

FROM Clientes c

JOIN Ventas v ON c.cedula = v.cedula\_cliente

GROUP BY c.nombre, v.medio\_pago;

# Conclusión

La gestión eficiente del inventario y la automatización de procesos son aspectos cruciales para el éxito de cualquier empresa, sin importar su tamaño. La microempresa RV Car Accesorios, ubicada en Ciudad Quesada, enfrenta importantes desafíos en la administración de su inventario y la mejora de su flujo de trabajo.

RV Car Accesorios al una amplia gama de accesorios. La empresa lucha por mantener su inventario actualizado y por gestionar los pedidos de manera efectiva, ya que aún depende de un sistema de control en papel y la memoria del propietario.

El objetivo de este proyecto es crear y utilizar una base de datos que ayude a “RV Car Accesorios” a realizar un seguimiento de sus productos, realizar facturas de manera automatizada.

El plan propuesto incluye la implementación de una base de datos que aborde los problemas de gestión de inventario y mejore la eficacia operativa de la empresa. Esto se logra a través de varias tablas diseñadas para almacenar información de manera organizada y eficiente.

La justificación y objetivos para la implementación de la base de datos incluye la necesidad de una gestión de inventario eficiente, control de acceso y seguridad, generación automatizada de facturas y análisis de datos y tendencias para la toma de decisiones estratégicas.

La solución propuesta se basa en una serie de tablas que almacenan información detallada sobre usuarios, clientes, proveedores, productos, ventas, facturas, entre otros. Estas tablas están diseñadas para permitir una gestión eficiente de la información y la generación de informes relevantes.

En resumen, la implementación de esta base de datos proporcionará a RV Car Accesorios las herramientas necesarias para mejorar su gestión de inventario, optimizar sus procesos y tomar decisiones comerciales más informadas. Con una base de datos centralizada y bien estructurada, la empresa estará mejor equipada para enfrentar los desafíos del mercado y alcanzar el éxito a largo plazo.

# **Bibliografia**

 Scribbr. (2024, March 1). *Free APA citation Generator | with Chrome Extension - Scribbr*. <https://www.scribbr.com/citation/generator/apa/>

Ramírez-Galvis, J. P. (2021). Plantilla de investigación con formato APA. *ResearchGate*. <https://doi.org/10.13140/RG.2.2.11331.94248>

Excel. (2022, November 13). *Listados con datos de ejemplo para descargar*. BUSCARV. <https://buscarv.com/plantillas/listados-con-datos-de-ejemplo-para-descargar/>

*diagrama de bases de datos - Bing*. (n.d.). Bing. <https://www.bing.com/images/search?view=detailV2&ccid=LFpCWrwv&id=D5CEC37353BF07541FE660C4DB1B45191BBA675A&thid=OIP.LFpCWrwv6Pp5fsMUeAKOMwHaFj&mediaurl=https%3a%2f%2fth.bing.com%2fth%2fid%2fR.2c5a425abc2fe8fa797ec31478028e33%3frik%3dWme6GxlFG9vEYA%26riu%3dhttp%253a%252f%252f3.bp.blogspot.com%252f-8H27D_eTUHA%252fUkYe6YlD_5I%252fAAAAAAAAA6M%252fG37qMlZjfiU%252fs1600%252fbdHotell.jpg%26ehk%3db47Alll8BhdFpoLO2wuqrlSJ3bFKXolqhZqdi9NWWlc%253d%26risl%3d%26pid%3dImgRaw%26r%3d0&exph=600&expw=800&q=diagrama+de+bases+de+datos&simid=608032284512488785&FORM=IRPRST&ck=4AA23A09887DCDF26C3A947F2959F34D&selectedIndex=5&itb=0&ajaxhist=0&ajaxserp=0>

Unknown. (n.d.). *1- Desarrollo de Aplicaciones con SqlServer y C#*. <https://rafaelturpoa.blogspot.com/2014/11/1-desarrollo-de-aplicaciones-con.html>