****

**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](#_Toc164795033)

[Introducción 3](#_Toc164795034)

[Planteamiento del problema y justificación 4](#_Toc164795035)

[Objetivos 5](#_Toc164795036)

[Objetivo general: 5](#_Toc164795037)

[Objetivos específicos: 5](#_Toc164795038)

[Descripción de la Solución 6](#_Toc164795039)

[Diagrama E/R 11](#_Toc164795040)

[Scripts 12](#_Toc164795041)

[Conclusión 13](#_Toc164795042)

[**Bibliografia** 14](#_Toc164795043)

# Introducción

Esta introducción contiene:

La importancia del trabajo

El origen del trabajo

Los antecedentes teóricos y prácticos (base científica del trabajo)

Los objetivos del trabajo

Los alcances del trabajo

Las limitaciones del trabajo

La metodología empleada en el trabajo

El significado que el estudio tiene en el avance del campo respectivo

La aplicación en el área investigada

NOTA

No confundir con el resumen

No incluir tablas ni resultados

En lo posible evitar citas de otros autores. Si lo hace, debe manejar citas y referencias de acuerdo con el manual de publicaciones de la APA.

La introducción debe tener una extensión de mínimo 2 páginas y máximo de 4 páginas.

# Planteamiento del problema y justificación

Debe contener:

Una descripción del problema de investigación de manera clara, detallada y no ambigua.

Las causas que originan el problema.

Las manifestaciones del problema.

El contraste con los límites o vacíos del conocimiento existente respecto al problema.

La importancia de solucionar este problema.

La delimitación del problema de investigación.

# Objetivos

## Objetivo general:

Corresponde al propósito fundamental del proyecto de investigación.

## Objetivos específicos:

Corresponden a los propósitos parciales del proyecto de investigación para lograr el propósito general.

Para tener en cuenta:

El propósito general debe estar en coherencia con el título del trabajo.

Los objetivos específicos deben estar en coherencia con el objetivo general.

Los objetivos (general y específicos) deben redactarse de manera concisa y detallada.

# Descripción de la Solución

Para la solución de el problema se crearon varias tablas que almacenan la información pertinente de forma ordenada y eficiente para el funcionamiento de la microempresa,a continuación la descripción de cada tabla y su función .

Usuarios:

* Almacena la información de los usuarios del sistema.
* Contiene detalles como cédula (identificación), nombre, apellidos, teléfono y correo electrónico de los usuarios.
* La cédula se utiliza como clave principal para identificar de manera única a cada usuario.

CREATE TABLE Usuarios (

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

nombre VARCHAR(50),

apellidos VARCHAR(50),

telefono VARCHAR(20),

correo VARCHAR(100)

);

Proveedores:

* Registra los proveedores de productos.
* Cada proveedor tiene un identificador único (ID), nombre, nombre del producto que provee, dirección, teléfono, correo electrónico y una descripción breve de los servicios o productos ofrecidos.
* Los proveedores se pueden contactar a través de los detalles de contacto almacenados en otras tablas.

-- 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)

);

Clientes:

* Almacena los datos de los clientes que realizan compras.
* Cada cliente tiene una cédula (identificación), nombre, teléfono, nombre de la empresa (si aplica) y correo electrónico.
* La cédula se utiliza como clave principal para identificar de manera única a cada cliente.

-- 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)

);

Direcciones Clientes:

* Guarda las direcciones asociadas a los clientes.
* Cada dirección tiene un identificador único, la cédula del cliente al que pertenece y la dirección en sí.
* Las direcciones están relacionadas con los clientes a través de la cédula del cliente.

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)

);

Direcciones Proveedores:

* Almacena las direcciones de los proveedores.
* Cada dirección tiene un identificador único, el ID del proveedor al que pertenece y la dirección en sí.
* Las direcciones están vinculadas a los proveedores a través del ID del proveedor.

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)

);

Contactos Clientes:

* Contiene los detalles de contacto de los clientes.
* Cada contacto tiene un identificador único, la cédula del cliente asociado, teléfono y correo electrónico.
* Los contactos están relacionados con los clientes a través de la cédula del cliente.

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)

);

Contactos Proveedores:

* Registra los contactos de los proveedores.
* Cada contacto tiene un identificador único, el ID del proveedor asociado, teléfono y correo electrónico.
* Los contactos están vinculados a los proveedores a través del ID del proveedor.

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)

);

Contactos Usuarios:

* Almacena los detalles de contacto de los usuarios.
* Cada contacto tiene un identificador único, la cédula del usuario asociado, teléfono y correo electrónico.
* Los contactos están relacionados con los usuarios a través de la cédula del usuario.

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)

);

Facturas:

* Registra las facturas generadas por las ventas.
* Cada factura tiene un número único de venta, fecha, monto, cédula del cliente y cédula del usuario.
* Las facturas están relacionadas con los clientes y los usuarios a través de sus cédulas respectivas.

-- 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)

);

Ventas:

* Guarda los registros de las ventas realizadas.
* Cada venta tiene un número único de venta, fecha, monto, medio de pago, cédula del cliente y cédula del usuario.
* Las ventas están relacionadas con los clientes y los usuarios a través de sus cédulas respectivas.

-- Ventas

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:

* Almacena la información de los productos disponibles para la venta.
* Cada producto tiene un identificador único, marca, tamaño, color, peso, precio, estilo y descripción.
* El ID del producto se utiliza como clave principal para identificar de manera única a cada producto.

-- 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)

);

Detalles Venta:

* Guarda los detalles de las ventas, normalizando la relación entre ventas y productos.
* Cada detalle tiene un identificador único, el número de venta al que pertenece, el ID del producto vendido, cantidad y precio.
* Los detalles están relacionados con las ventas y los productos a través de sus números de venta y IDs de producto.

-- DetallesVenta

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)

);

categorías Productos:

* Registra las categorías a las que pertenecen los productos.
* Cada categoría tiene un identificador único, nombre y descripción.
* El ID de la categoría se utiliza como clave principal para identificar de manera única a cada categoría.

-- CategoriasProductos

CREATE TABLE CategoriasProductos (

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

nombre VARCHAR(100),

descripcion VARCHAR(80)

);

Producto Categoria:

* Establece la relación entre productos y categorías.
* Cada registro contiene el ID del producto y el ID de la categoría a la que pertenece.
* Este esquema permite que un producto pueda pertenecer a varias categorías y viceversa.

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)

);

¿

# 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

# 

# Conclusión

# 

# **Bibliografia**

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