Seguros Confianza



***https://github.com/Abentfork/Proyecto-final-BBDD-Adrian-Ramos-Espinosa***

**Nome Alumno/a:**

***Adrian Ramos Espinosa***

**Curso: *1º DAM* Materia:** ***Bases de Datos – Proyecto Final 24/25***

Contido

[1. Introducción 2](#_Toc197794426)

[2. Descripción del Problema / Requisitos 2](#_Toc197794427)

[3. Modelo Conceptual 2](#_Toc197794428)

[4. Modelo Relacional 2](#_Toc197794429)

[5. Proceso de Normalización 2](#_Toc197794430)

[6. Script de Creación de la Base de Datos 2](#_Toc197794431)

[7. Carga de Datos Inicial 2](#_Toc197794432)

[8. Funciones y Procedimientos Almacenados 2](#_Toc197794433)

[9. Triggers 2](#_Toc197794434)

[10. Consultas SQL 2](#_Toc197794435)

[11. Casos de Prueba y Simulación 2](#_Toc197794436)

[12. Resultados y Verificación 2](#_Toc197794437)

[13. Capturas de Pantalla (opcional) 2](#_Toc197794438)

[14. Conclusiones y Mejoras Futuras 2](#_Toc197794439)

[15. Enlace al Repositorio en GitHub 3](#_Toc197794440)

# Introducción

En este trabajo una empresa aseguradora nos contacta porque necesita una base de datos para

Gestionar las polizas de seguro

# Descripción del Problema / Requisitos

Se tiene una reunion con el cliente en la cual nos explica que quiere una base de datos que ayude a gestionar el negocio

Nos dice que las principales caracteristicas que precisa la base de datos son

-La gestion de clientes

-La gestion de las polizas

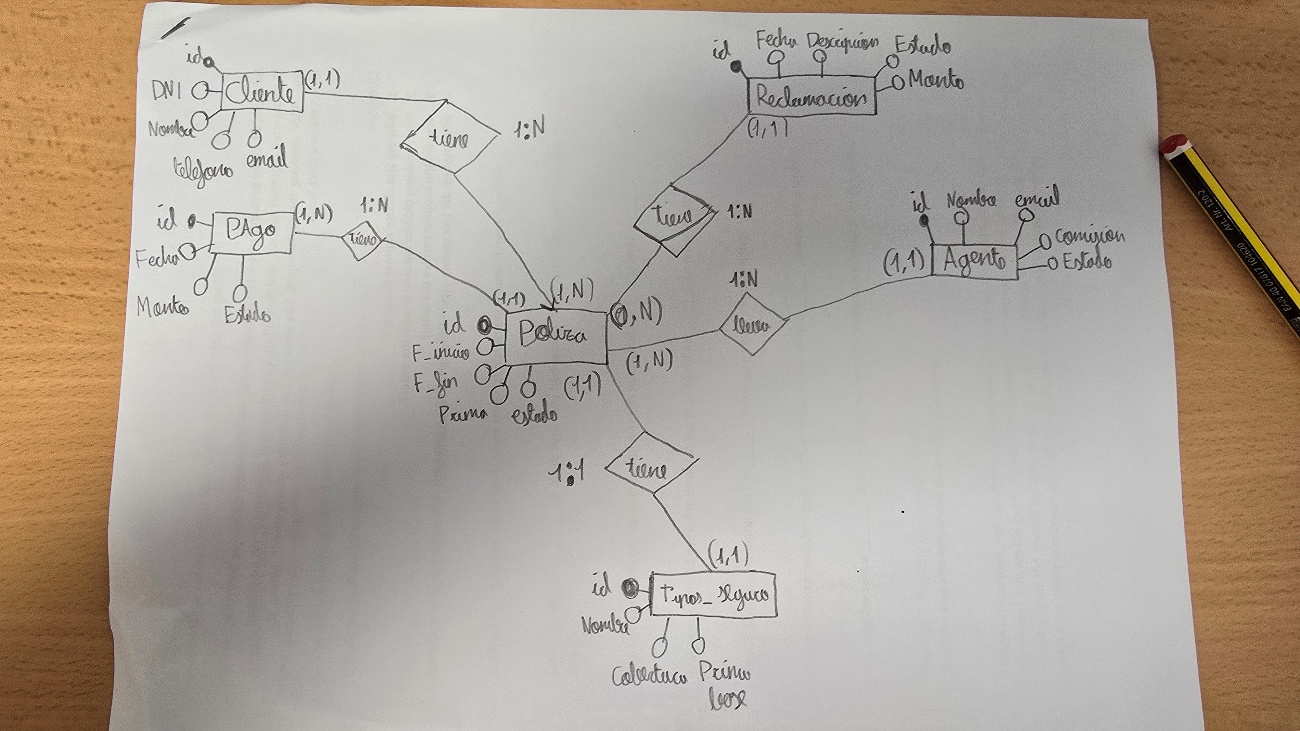
-La gestion de las reclamaciones de las polizas

-La gestion de los agentes y corredores de seguros

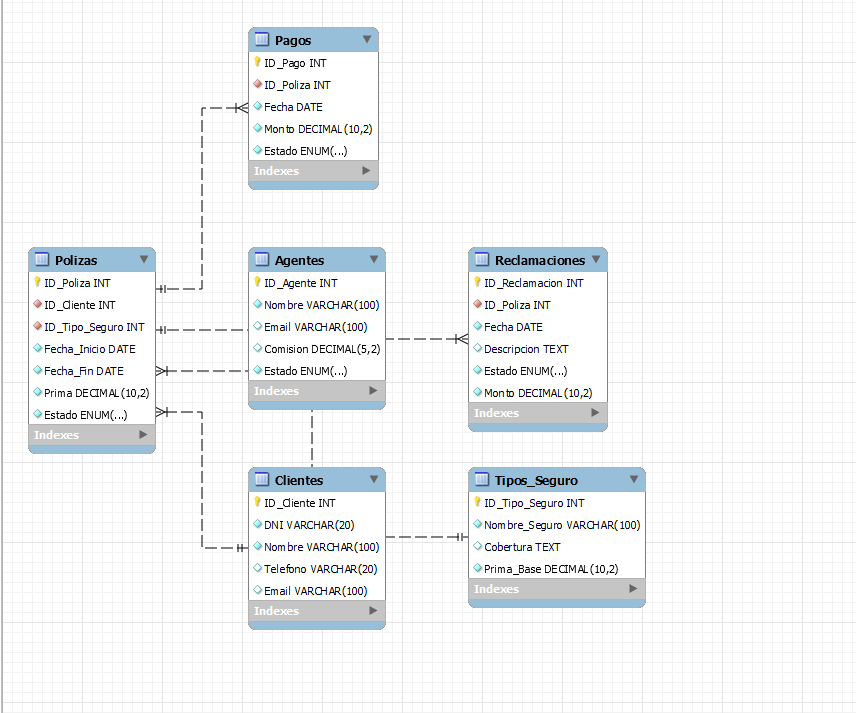
- La gestion financiera respecto a pagos y comisiones

# Modelo Conceptual

El modelo conceptual se realiza a partir de las tablas necesarias y es el siguiente



# Modelo Relacional



# Proceso de Normalización

Primero se separo cada necesidad en una tabla es decir los clientes, los agentes, los pagos etc

Despues se revisaron los datos para evitar repeticiones,

Se utilizaron claves foraneas para conectar las trablas sin duplicar informacion

Y me asegure que en cada tabla solo hubiera un tipo de dato

# Script de Creación de la Base de Datos

Para realizar el script de base de datos se partio de la informacion que el cliente me proporciono para realizar las tablas y campos y quedo tal que asi  
  
-- Crear base de datos

CREATE DATABASE IF NOT EXISTS aseguradora;

USE aseguradora;

# -- Tabla: CLIENTES

# CREATE TABLE Clientes (

# ID\_Cliente INT AUTO\_INCREMENT PRIMARY KEY,

# DNI VARCHAR(20) NOT NULL UNIQUE,

# Nombre VARCHAR(100) NOT NULL,

# Telefono VARCHAR(20),

# Email VARCHAR(100)

# );

# -- Tabla: TIPOS\_SEGURO

# CREATE TABLE Tipos\_Seguro (

# ID\_Tipo\_Seguro INT AUTO\_INCREMENT PRIMARY KEY,

# Nombre\_Seguro VARCHAR(100) NOT NULL,

# Cobertura TEXT,

# Prima\_Base DECIMAL(10, 2) NOT NULL

# );

# -- Tabla: POLIZAS

# CREATE TABLE Polizas (

# ID\_Poliza INT AUTO\_INCREMENT PRIMARY KEY,

# ID\_Cliente INT NOT NULL,

# ID\_Tipo\_Seguro INT NOT NULL,

# Fecha\_Inicio DATE NOT NULL,

# Fecha\_Fin DATE NOT NULL,

# Prima DECIMAL(10, 2) NOT NULL,

# Estado VARCHAR(20) NOT NULL DEFAULT 'Activa',

# FOREIGN KEY (ID\_Cliente) REFERENCES Clientes(ID\_Cliente),

# FOREIGN KEY (ID\_Tipo\_Seguro) REFERENCES Tipos\_Seguro(ID\_Tipo\_Seguro)

# );

# -- Tabla: RECLAMACIONES

# CREATE TABLE Reclamaciones (

# ID\_Reclamacion INT AUTO\_INCREMENT PRIMARY KEY,

# ID\_Poliza INT NOT NULL,

# Fecha DATE NOT NULL,

# Descripcion TEXT,

# Estado VARCHAR(20) NOT NULL DEFAULT 'Pendiente',

# Monto DECIMAL(10, 2) NOT NULL,

# FOREIGN KEY (ID\_Poliza) REFERENCES Polizas(ID\_Poliza)

# );

# -- Tabla: AGENTES

# CREATE TABLE Agentes (

# ID\_Agente INT AUTO\_INCREMENT PRIMARY KEY,

# Nombre VARCHAR(100) NOT NULL,

# Email VARCHAR(100),

# Comision DECIMAL(5, 2), -- Porcentaje: ej. 15.25%

# Estado VARCHAR(20) NOT NULL DEFAULT 'Activo'

# );

# -- Tabla: PAGOS

# CREATE TABLE Pagos (

# ID\_Pago INT AUTO\_INCREMENT PRIMARY KEY,

# ID\_Poliza INT NOT NULL,

# Fecha DATE NOT NULL,

# Monto DECIMAL(10, 2) NOT NULL,

# Estado VARCHAR(20) NOT NULL DEFAULT 'Pendiente',

# FOREIGN KEY (ID\_Poliza) REFERENCES Polizas(ID\_Poliza)

# Carga de Datos Inicial

Para la carga de datos voy a usar chatgt para que me de los datos

USE aseguradora;

-- 1. CLIENTES

INSERT INTO Clientes (ID\_Cliente, DNI, Nombre, Telefono, Email)

VALUES

(1, '12345678A', 'Juan Pérez', '600123456', 'juanp@gmail.com'),

(2, '23456789B', 'María López', '600234567', 'mlopez@gmail.com'),

(3, '34567890C', 'Carlos Ruiz', '600345678', 'cruiz@gmail.com');

-- 2. TIPOS\_SEGURO

INSERT INTO Tipos\_Seguro (ID\_Tipo\_Seguro, Nombre\_Seguro, Cobertura, Prima\_Base)

VALUES

(1, 'Hogar', 'Daños, incendios, robos', 300.00),

(2, 'Coche', 'Accidentes, robo, terceros', 450.00),

(3, 'Vida', 'Fallecimiento, invalidez', 600.00);

-- 3. POLIZAS

INSERT INTO Polizas (ID\_Poliza, ID\_Cliente, ID\_Tipo\_Seguro, Fecha\_Inicio, Fecha\_Fin, Prima, Estado)

VALUES

(1, 1, 1, '2024-01-01', '2025-01-01', 320.00, 'Activa'),

(2, 2, 2, '2024-03-01', '2025-03-01', 470.00, 'Activa'),

(3, 3, 3, '2023-06-15', '2024-06-15', 610.00, 'Vencida');

-- 4. RECLAMACIONES

INSERT INTO Reclamaciones (ID\_Reclamacion, ID\_Poliza, Fecha, Descripcion, Estado, Monto)

VALUES

(1, 1, '2024-02-10', 'Daño por agua', 'Pendiente', 1200.00),

(2, 2, '2024-04-20', 'Accidente de tráfico', 'Procesado', 2500.00),

(3, 3, '2024-01-05', 'Cobro por invalidez', 'Aprobado', 10000.00);

-- 5. AGENTES

INSERT INTO Agentes (ID\_Agente, Nombre, Email, Comision, Estado)

VALUES

(1, 'Luis Torres', 'ltorres@aseg.com', 5.50, 'Activo'),

(2, 'Ana Gómez', 'agomez@aseg.com', 6.00, 'Activo'),

(3, 'Pedro Díaz', 'pdiaz@aseg.com', 4.75, 'Inactivo');

-- 6. PAGOS

INSERT INTO Pagos (ID\_Pago, ID\_Poliza, Fecha, Monto, Estado)

VALUES

(1, 1, '2024-01-05', 320.00, 'Pagado'),

(2, 2, '2024-03-10', 470.00, 'Pagado'),

(3, 3, '2023-06-20', 610.00, 'Pendiente');

# Funciones y Procedimientos Almacenados

Funciones se haran 2 : una para calcular la duracion de la poliza en dias y otra para el total que se ha pagado de una poliza

Procedimientos tambien 2: Uno para crear una nueva reclamacion de poliza y otro para cancelar una poliza

# Triggers

El primer trigger : antes de insertar en la tabla de pagos se asegura de que el monto no sea negativo

El segundo trigger: Cada vez que se hace una nueva reclamacion cambia el estado de la poliza asociada a “En revision”

# Consultas SQL

Describe aquí...

# Casos de Prueba y Simulación

Describe aquí...

# Resultados y Verificación

Describe aquí...

# Capturas de Pantalla (opcional)

Describe aquí...

# Conclusiones y Mejoras Futuras

Describe aquí...

# Enlace al Repositorio en GitHub

Describe aquí...