

Universidad Tecnológica Boliviana

La paz, 12 de diciembre de 2023

**Proyecto de base de datos I: Sistema de gestión de datos para Importadora Genolab**

Asignatura : Base de datos I

Docente : Ing. Angela Guadalupe Santos Quisbert

Estudiante : Fabio Camacho Encinas

Carrera : Ingeniería de sistemas

Facultad : Electrónica y Sistemas

Contenido

[1. Introducción 3](#_Toc153299642)

[2. Planteamiento del problema 4](#_Toc153299643)

[3. Análisis del negocio 5](#_Toc153299644)

[3.1. Estructura y operaciones del negocio 5](#_Toc153299645)

[3.2. Requerimientos. 7](#_Toc153299646)

[3.2.1 Requerimientos funcionales 7](#_Toc153299647)

[3.2.1 Requerimientos no funcionales 8](#_Toc153299648)

[4. Estructura y desarrollo del sistema. 8](#_Toc153299649)

[4.1. Herramientas 8](#_Toc153299650)

[4.2. Arquitectura del sistema 10](#_Toc153299651)

[4.3. Base de datos 11](#_Toc153299652)

[4.3.1 Modelo lógico 11](#_Toc153299653)

[4.3.2 Modelo lógico relacional 12](#_Toc153299654)

[4.4 Capa de presentación 16](#_Toc153299655)

[4.4.1 Maquetado 16](#_Toc153299656)

# 1. Introducción

El presente proyecto consiste en el diseño y desarrollo de un sistema de gestión de información destinado a implementarse en las operaciones de una importadora de medicamentos y suministros de salud denominada Genolab, que realizara sus actividades económicas dentro del territorio nacional.

Este informe describe brevemente el análisis previo de la empresa, las entidades que intervienen en las operaciones y el diseño de las diferentes estructuras del sistema.

Marco teórico

**Base de datos:** Una base de datos es una recopilación de datos sistemática y almacenada electrónicamente. Puede contener cualquier tipo de datos, incluidos palabras, números, imágenes, vídeos y archivos. Puede usar un software denominado sistema de administración de bases de datos (DBMS) para almacenar, recuperar y editar datos. En los sistemas informáticos, la palabra base de datos también puede referirse a cualquier DBMS, al sistema de base de datos o a una aplicación asociada con la base de datos.

Fuente: <https://aws.amazon.com/es/what-is/database/>

**SGBD:** Un Sistema Gestor de Base de Datos (SGBD) o DGBA (Data Base Management System) es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos Los gestores de base de datos o gestores de datos hacen posible admnistrar todo acceso a la base de datos ya que tienen el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones.

**Base de datos relacional:** Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas. En una base de datos relacional, cada fila en una tabla es un registro con una ID única, llamada clave. Las columnas de la tabla contienen los atributos de los datos y cada registro suele tener un valor para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos.

Fuente : https://www.oracle.com/ar/database/what-is-a-relational-database/

# 2. Planteamiento del problema

Se han identificado requisitos internos y externos que sugieren la implementación de un sistema informático de gestión de información sobre un sistema tradicional:

Requisitos internos:

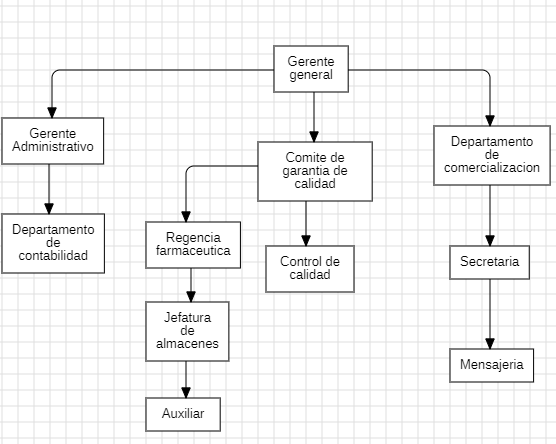
* Manejo de una gran cantidad de registros (solo en el Kardex de productos, se estima que se tendrá que procesar al menos 50 registros por 180 artículos y cada artículo con al menos 5 lotes)
* Eficiencia en la gestión de registros contables.
* Agilidad en las operaciones comerciales.
* Respaldo solido de toda la información que interviene en las operaciones comerciales

Requisitos externos:

* Dentro del territorio nacional, el Servicio de Impuestos Nacionales exige a las empresas con las características de Genolab, cuenten con un sistema electrónico de facturación en línea.
* Las empresas que comercializan medicamentos y suministros de salud son supervisadas por la Agencia Estatal de Medicamentos y Tecnologías en salud. Esta agencia exige una gestión detallada y precisa de cada transacción realizada con artículos a la salud del público en general.

# 3. Análisis del negocio

# 3.1. Estructura y operaciones del negocio

La estructura organizativa de Genolab es la siguiente:  


Existen 3 áreas principales: Gerencia y contabilidad, Área de comercialización y Logística y distribución.

Posterior a entrevistas dentro de las diferentes áreas de la empresa, se ha recopilado información del flujo de las operaciones habituales:

* La empresa comercializa medicamentos y suministros de salud.
* Los artículos son importados de laboratorios del exterior.
* Cada artículo tiene una descripción, concentración y forma farmacéutica. Por ejemplo: “Diclofenaco 50 mg Comprimido”
* Los artículos son importados en lotes y puede haber varios lotes del mismo articulo siendo comercializados al mismo tiempo.
* Cada lote tiene fecha de fabricación, expiración y código asignado.
* Tras un análisis de mercado, la jefatura del Departamento de comercialización requiere importar un nuevo lote de artículos, y hace un pedido a determinado laboratorio, indicando la cantidad que se desea importar.
* Logística se encarga de la recepción del lote de producto en almacenes de recepción.
* Regencia farmacéutica hace un análisis bioquímico de las muestras del lote recibido.
* Si los análisis están dentro de parámetros establecidos, regencia farmacéutica registra el producto como “apto para comercialización” y emite un informe al Departamento de comercialización.
* Solo y solo Regencia puede autorizar el registro de un nuevo lote de artículos.
* Tras un análisis de mercado, el Departamento de comercialización determina el precio unitario para el lote de producto.
* La empresa vende productos únicamente a distribuidores autorizados. Por lo tanto, el Departamento de comercialización vende los artículos a clientes pre registrados, y en los registros se tiene la siguiente información de cada cliente: Nombre, tipo de documento, numero de documento, teléfono e e-mail.
* Cada cliente puede operar en varias sucursales y en varias regiones del país, así que se registran direcciones designadas a determinado cliente.
* Cuando un cliente desea adquirir un producto, se comunica con Departamento de comercialización, indicando la cantidad y el producto que desea adquirir.
* El ejecutivo del Departamento de comercialización entonces genera una orden de venta, indicando la fecha, hora, nombre de cliente y dirección en la cual desea recibir los productos.
* El ejecutivo de ventas agrega a la orden de venta cada producto que desea adquirir el cliente, indicando el lote y la cantidad, precio unitario y subtotal, adquiriendo también el importe total de la venta.
* Cuando el cliente realiza el pago, el ejecutivo de ventas genera una factura en base a la orden de venta.
* Una vez realizada la orden de venta y factura, el ejecutivo de ventas extiende la orden de venta a Logística y distribución.
* Jefatura de almacenes asigna a un auxiliar a preparar el pedido, y coordina el envió con una empresa de transporte de carga y flete.
* Periódicamente, Contabilidad realiza inventario de los productos, operación que es coordinada con Logística.

# 3.2. Requerimientos.

# 3.2.1 Requerimientos funcionales

* El sistema debe funcionar con roles definidos para el acceso a determinadas operaciones. Ejemplo: Ventas, Administración, Regencia Farmaceutica, Almacenes
* El sistema debe servir para la gestión de los productos, permitiendo indicar la forma farmacéutica, el laboratorio de procedencia, fechas de fabricación expiración, y cantidad existente.
* El sistema debe permitir registrar clientes, y las sucursales donde se harán las entregas a los clientes.
* También debe servir para gestionar operaciones de venta, donde en cada venta se pueden asignar un lista de productos que el cliente desea adquirir  
    
  Mediante estos requerimientos funcionales, se determinó la creación de los siguientes módulos.
* **Módulo de inscripción de artículos:** Una interfaz gráfica para inscribir nuevos artículos.
* **Módulo de inscripción de lotes:** Una interfaz gráfica para inscribir lotes.
* **Módulo de orden de venta:**  Un módulo para agregar los productos y cantidades que el cliente desea adquirir.
* **Modulo para inscripción de clientes:** Permite agregar nuevos clientes a la base de datos.
* **Módulo de direcciones:** Permite agregar nuevas direcciones relacionadas a clientes.
* **Módulo de facturación:** Genera facturas basadas en un detalle de los pedidos.
* **Módulo de creación de usuario y administración de roles:** Un módulo donde el gerente administrativo pueda crear nuevos roles, que permiten o restringen el acceso a determinados módulos.
* **Módulo de inicio de sesión:** Donde el usuario pueda ingresar sus credenciales asignadas de acuerdo a su rol y conectarse al servidor para usar los módulos designados.

# 3.2.1 Requerimientos no funcionales

* El sistema debe ser capaz de manejar 10 usuarios concurrentes.
* El sistema debe implementar medidas de seguridad para proteger la información confidencial de los clientes, como datos personales y detalles financieros.
* El sistema debe ser capaz de manejar de manera eficiente y sin demoras significativas un volumen de transacciones esperado durante las horas pico de ventas.
* Debe mantener un registro detallado de todas las transacciones realizadas, incluyendo la modificación de datos y el acceso al sistema por parte de los usuarios.
* Debe existir un sistema regular de respaldo de datos y un plan de recuperación ante desastres para minimizar la pérdida de información en caso de fallos del sistema.
* La interfaz de usuario debe ser intuitiva y fácil de usar, minimizando la necesidad de capacitación extensa para nuevos usuarios.

# 4. Estructura y desarrollo del sistema.

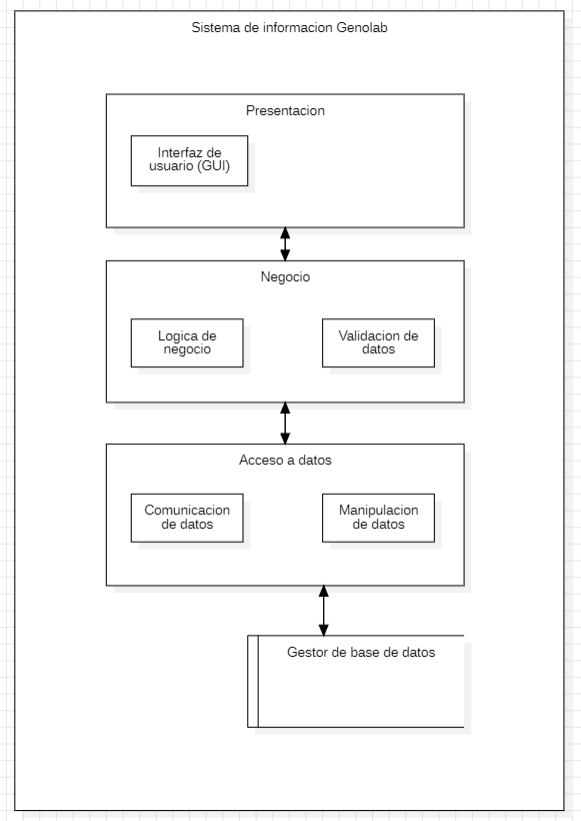
# 4.1. Herramientas

Las herramientas utilizadas para el desarrollo del sistema son:

* Java.- Lógica y diseño de interfaz gráfica de usuario.
* MySQL.- Gestión y diseño de base de datos.

# 4.2. Arquitectura del sistema

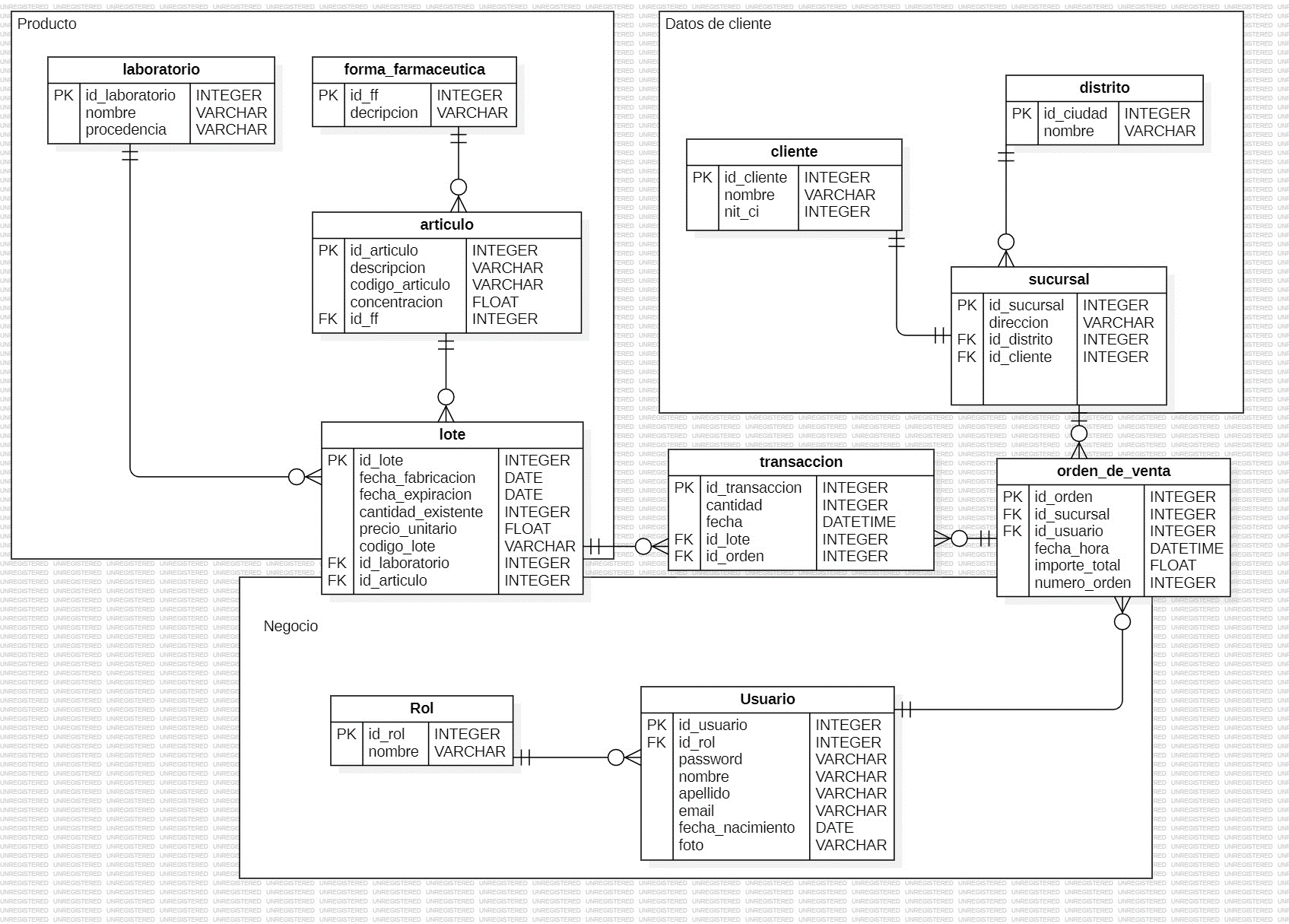
Para el proyecto, se optará por la arquitectura de desarrollo en capas, donde cada capa únicamente se puede comunicar con las capas adyacentes. Esta estructura es modular, lo cual aporta en facilidad para el mantenimiento del sistema, migración de datos, etc.



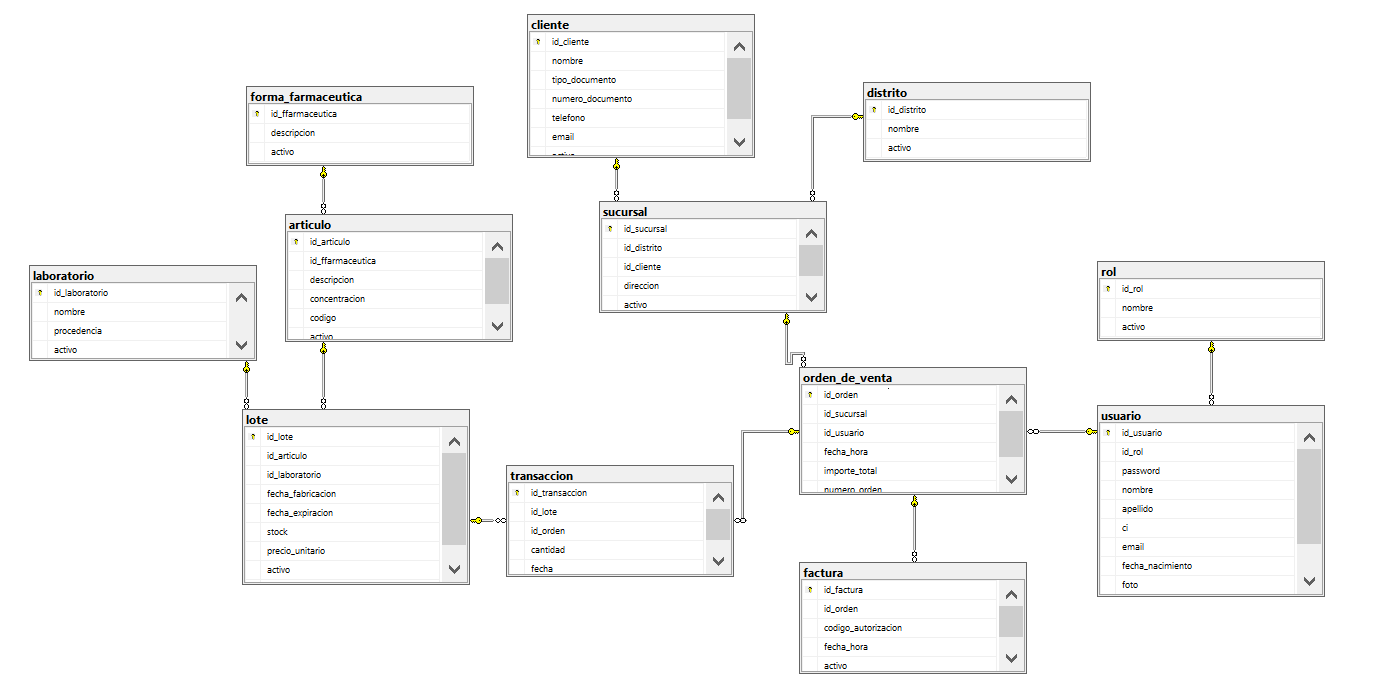
# 4.3. Base de datos

# 4.3.1 Modelo lógico

Mediante el análisis de la estructura y las operaciones de la empresa, se han identificado estas entidades y sus relaciones para diseñar el siguiente modelo lógico:



# 4.3.2 Modelo lógico relacional



4.3.3 Consulta de creación de base de datos

-- Crear la base de datos

CREATE DATABASE bd\_genolab;

-- Usar la base de datos

USE bd\_genolab;

-- Crear la tabla de forma\_farmaceutica

CREATE TABLE forma\_farmaceutica (

id\_ffarmaceutica INT PRIMARY KEY IDENTITY(1,1),

descripcion VARCHAR(40) NOT NULL,

activo BIT NOT NULL

);

-- Crear la tabla laboratorio

CREATE TABLE laboratorio (

id\_laboratorio INT PRIMARY KEY IDENTITY(1,1),

nombre VARCHAR(75) NOT NULL,

procedencia VARCHAR(75) NOT NULL,

activo BIT NOT NULL

);

-- Crear la tabla distrito

CREATE TABLE distrito (

id\_distrito INT PRIMARY KEY IDENTITY(1,1),

nombre VARCHAR(75) NOT NULL,

activo BIT NOT NULL

);

-- Crear la tabla cliente

CREATE TABLE cliente (

id\_cliente INT PRIMARY KEY IDENTITY(1,1),

nombre VARCHAR(100) NOT NULL,

tipo\_documento VARCHAR(45) NOT NULL,

numero\_documento VARCHAR(45) NOT NULL,

telefono VARCHAR(45) NOT NULL,

email VARCHAR(45),

activo BIT NOT NULL

);

-- Crear la tabla articulo

CREATE TABLE articulo (

id\_articulo INT PRIMARY KEY IDENTITY(1,1),

id\_ffarmaceutica INT NOT NULL,

descripcion VARCHAR(100) NOT NULL,

concentracion FLOAT,

codigo VARCHAR(15) NOT NULL,

activo BIT NOT NULL,

CONSTRAINT fk\_articulo\_ffarmaceutica FOREIGN KEY (id\_ffarmaceutica) REFERENCES forma\_farmaceutica (id\_ffarmaceutica)

);

-- Crear la tabla lote

CREATE TABLE lote (

id\_lote INT PRIMARY KEY IDENTITY(1,1),

id\_articulo INT NOT NULL,

id\_laboratorio INT NOT NULL,

fecha\_fabricacion DATE NOT NULL,

fecha\_expiracion DATE NOT NULL,

stock INT NOT NULL,

precio\_unitario DECIMAL(5,2) NOT NULL,

activo BIT NOT NULL,

codigo VARCHAR(20) NOT NULL,

CONSTRAINT fk\_lote\_articulo FOREIGN KEY (id\_articulo) REFERENCES articulo (id\_articulo),

CONSTRAINT fk\_lote\_laboratorio FOREIGN KEY (id\_laboratorio) REFERENCES laboratorio (id\_laboratorio)

);

-- Crear la tabla sucursal

CREATE TABLE sucursal (

id\_sucursal INT PRIMARY KEY IDENTITY(1,1),

id\_distrito INT NOT NULL,

id\_cliente INT NOT NULL,

direccion varchar(200),

activo BIT NOT NULL,

CONSTRAINT fk\_sucursal\_cliente FOREIGN KEY (id\_cliente) REFERENCES cliente (id\_cliente),

CONSTRAINT fk\_sucursal\_distrito FOREIGN KEY (id\_distrito) REFERENCES distrito (id\_distrito)

);

-- Crear la tabla rol

CREATE TABLE rol (

id\_rol INT PRIMARY KEY IDENTITY(1,1),

nombre VARCHAR(60) NOT NULL,

activo BIT NOT NULL

);

-- Crear la tabla usuario

CREATE TABLE usuario (

id\_usuario INT PRIMARY KEY IDENTITY(1,1),

id\_rol INT NOT NULL,

password VARCHAR(64) NOT NULL,

nombre VARCHAR(100) NOT NULL,

apellido VARCHAR(100) NOT NULL,

ci VARCHAR(20) NOT NULL,

email VARCHAR(50),

fecha\_nacimiento DATE,

foto VARCHAR(70),

activo BIT NOT NULL,

CONSTRAINT fk\_usuario\_rol FOREIGN KEY (id\_rol) REFERENCES rol (id\_rol)

);

-- Crear la tabla orden\_de\_venta

CREATE TABLE orden\_de\_venta (

id\_orden INT PRIMARY KEY IDENTITY(1,1),

id\_sucursal INT NOT NULL,

id\_usuario INT NOT NULL,

fecha\_hora DATETIME NOT NULL,

importe\_total DECIMAL(10,2),

numero\_orden INT,

CONSTRAINT fk\_orden\_sucursal FOREIGN KEY (id\_sucursal) REFERENCES sucursal (id\_sucursal),

CONSTRAINT fk\_orden\_usuario FOREIGN KEY (id\_usuario) REFERENCES usuario (id\_usuario)

);

-- Crear la tabla factura

CREATE TABLE factura (

id\_factura INT PRIMARY KEY IDENTITY(1,1),

id\_orden INT NOT NULL,

codigo\_autorizacion VARCHAR(45) NOT NULL,

fecha\_hora DATETIME NOT NULL,

activo BIT NOT NULL,

CONSTRAINT fk\_factura\_orden FOREIGN KEY (id\_orden) REFERENCES orden\_de\_venta (id\_orden)

);

-- Crear la tabla transaccion

CREATE TABLE transaccion (

id\_transaccion INT PRIMARY KEY IDENTITY(1,1),

id\_lote INT NOT NULL,

id\_orden INT NOT NULL,

cantidad INT NOT NULL,

fecha DATETIME NOT NULL,

CONSTRAINT fk\_transaccion\_lote FOREIGN KEY (id\_lote) REFERENCES lote (id\_lote),

CONSTRAINT fk\_transaccion\_orden FOREIGN KEY (id\_orden) REFERENCES orden\_de\_venta (id\_orden)

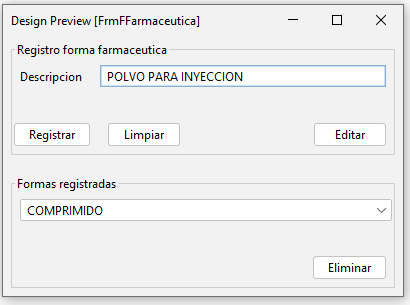
);

# 4.4 Capa de presentación

# 4.4.1 Maquetado

De acuerdo a los requerimientos funcionales, se han maquetado los siguientes módulos con algunos ejemplos de sus procedimientos almacenados.

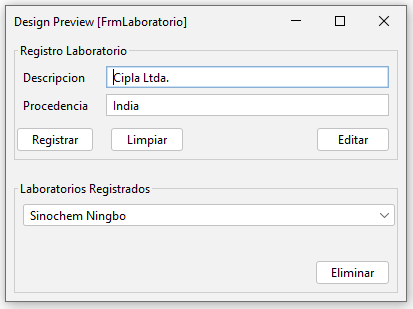
**Módulo de forma farmacéutica**



INSERT INTO `bd\_genolab`.`forma\_farmaceutica` (`descripcion`) VALUES

('Comprimido'),

**Módulo de laboratorio**



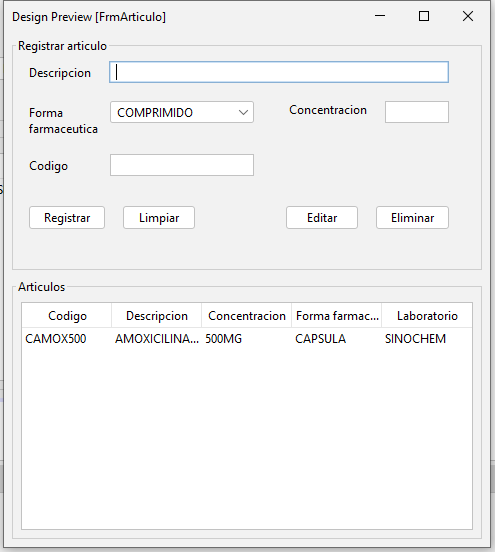
INSERT INTO `bd\_genolab`.`laboratorio` (`nombre`, `procedencia`, `activo`) VALUES

('Sinochem', 'China', 1),

('Ningbo', 'China', 1),

('Laboratorios Chile', 'Chile', 1);

**Modulo de articulo**



INSERT INTO `bd\_genolab`.`articulo` (`id\_ffarmaceutica`, `descripcion`, `concentracion`, `codigo`, `activo`) VALUES

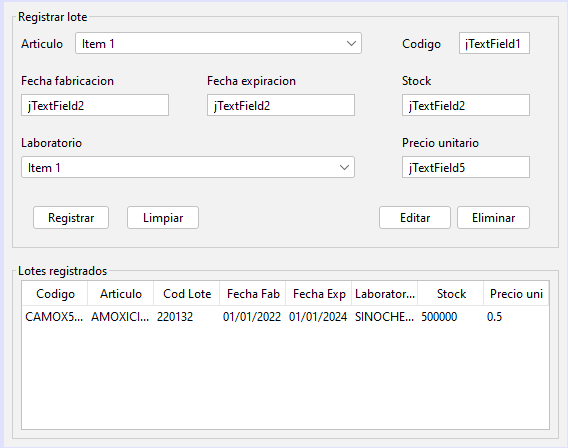
(1, 'Amoxicilina 500mg', 500, 'AMOX001', 1),

(2, 'Complejo B 2ml', 2, 'CB001', 1),

(1, 'Paracetamol 500mg', 500, 'PARA001', 1),

(3, 'Vitamina C 5ml', 5, 'VITC001', 1);

**Modulo Lote**



INSERT INTO `bd\_genolab`.`lote` (`id\_articulo`, `id\_laboratorio`, `fecha\_fabricacion`, `fecha\_expiracion`, `stock`, `precio\_unitario`, `activo`, `codigo`) VALUES

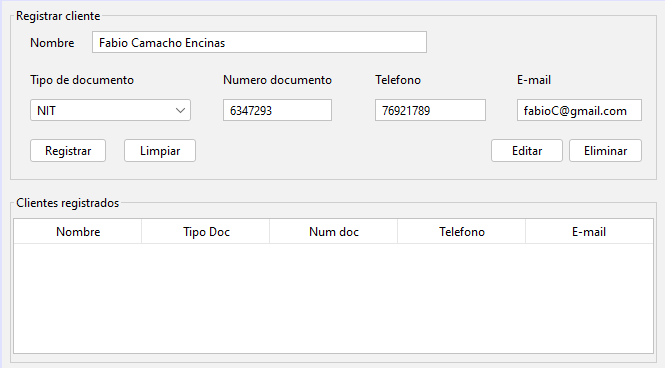
(1, 1, '2023-11-01', '2024-11-01', 500000, 0.5, 1, 'LOTE001'),

(2, 2, '2023-11-01', '2024-11-01', 500000, 0.4, 1, 'LOTE002'),

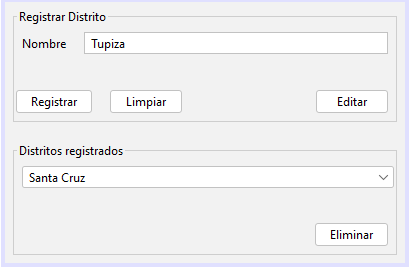
**Modulo Cliente**

INSERT INTO `bd\_genolab`.`cliente` (`nombre`, `tipo\_documento`, `numero\_documento`, `telefono`, `email`, `activo`) VALUES

('Juan Ramos', 'NIT', '123456789', '555-1234', 'juan.ramos@example.com', 1),



**Modulo Distrito**



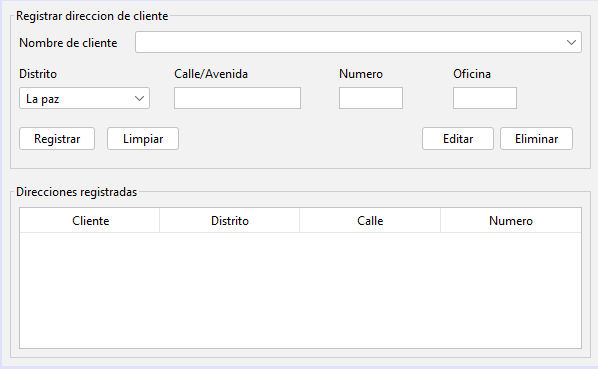
INSERT INTO `bd\_genolab`.`distrito` (`nombre`) VALUES

('Cochabamba'),

('La Paz'),

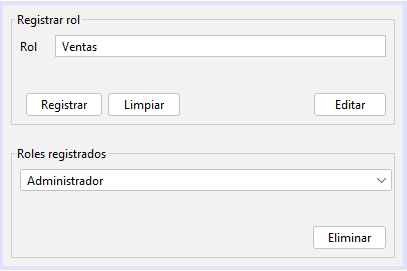
('Santa Cruz');

**Modulo Dirección**



INSERT INTO `bd\_genolab`.`direccion` (`id\_distrito`, `id\_cliente`, `calle`, `numero`, `oficina`, `activo`) VALUES

(1, 1, 'Calle A', '123', 'Oficina 1', 1),

**Modulo Rol**  


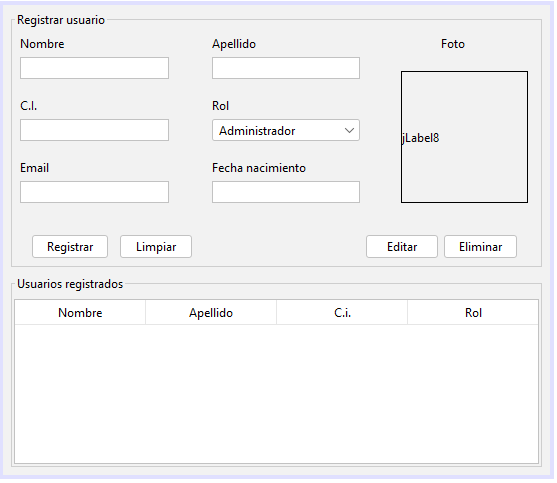
INSERT INTO `bd\_genolab`.`rol` (`id\_empresa`, `nombre`, `activo`) VALUES

(1, 'Administrador', 1),

(1, 'Ventas', 1),

(1, 'Regencia Farmaceutica', 1),

**Modulo Usuario**

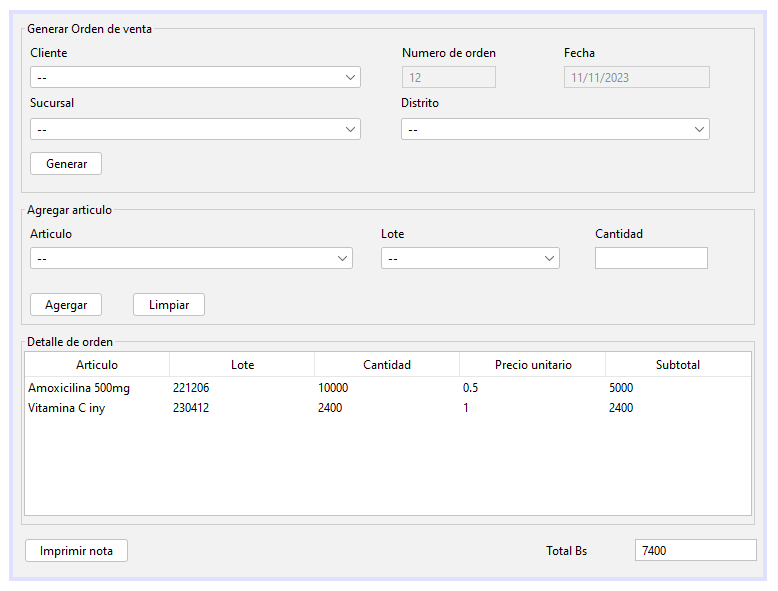


INSERT INTO `bd\_genolab`.`usuario` (`id\_rol`, `password`, `nombre`, `apellido`, `ci`, `email`, `fecha\_nacimiento`, `activo`) VALUES

(1, 'password\_admin', 'Carlos', 'López', '123456', 'carlos.lopez@example.com', '1990-01-01', 1),

(2, 'password\_ventas', 'Sofía', 'González', '789012', 'sofia.gonzalez@example.com', '1992-05-15', 1)

**Modulo orden de venta**



INSERT INTO `bd\_genolab`.`orden\_de\_venta` (`id\_cliente`, `id\_usuario`, `fecha\_hora`) VALUES

(1, 2, NOW()),

(2, 2, NOW()),

(3, 2, NOW());

-- Insertar transacciones para la Orden de Venta 1

INSERT INTO `bd\_genolab`.`transaccion` (`id\_lote`, `id\_orden`, `cantidad`, `fecha`) VALUES

(1, 1, 6000, NOW()),

(2, 1, 2000, NOW());

-- INICIO OPERACION VENTA --

UPDATE `bd\_genolab`.`orden\_de\_venta` AS o

SET o.`importe\_total` = (

SELECT SUM(t.`cantidad` \* l.`precio\_unitario`)

FROM `bd\_genolab`.`transaccion` AS t

INNER JOIN `bd\_genolab`.`lote` AS l ON t.`id\_lote` = l.`id\_lote`

WHERE t.`id\_orden` = o.`id\_orden`

)

WHERE o.`id\_orden` IN (1, 2, 3);

UPDATE `bd\_genolab`.`lote` AS l

SET l.`stock` = l.`stock` - (

SELECT SUM(t.`cantidad`)

FROM `bd\_genolab`.`transaccion` AS t

WHERE t.`id\_lote` = l.`id\_lote`

)

WHERE l.`id\_lote` IN (1, 2, 3, 4);