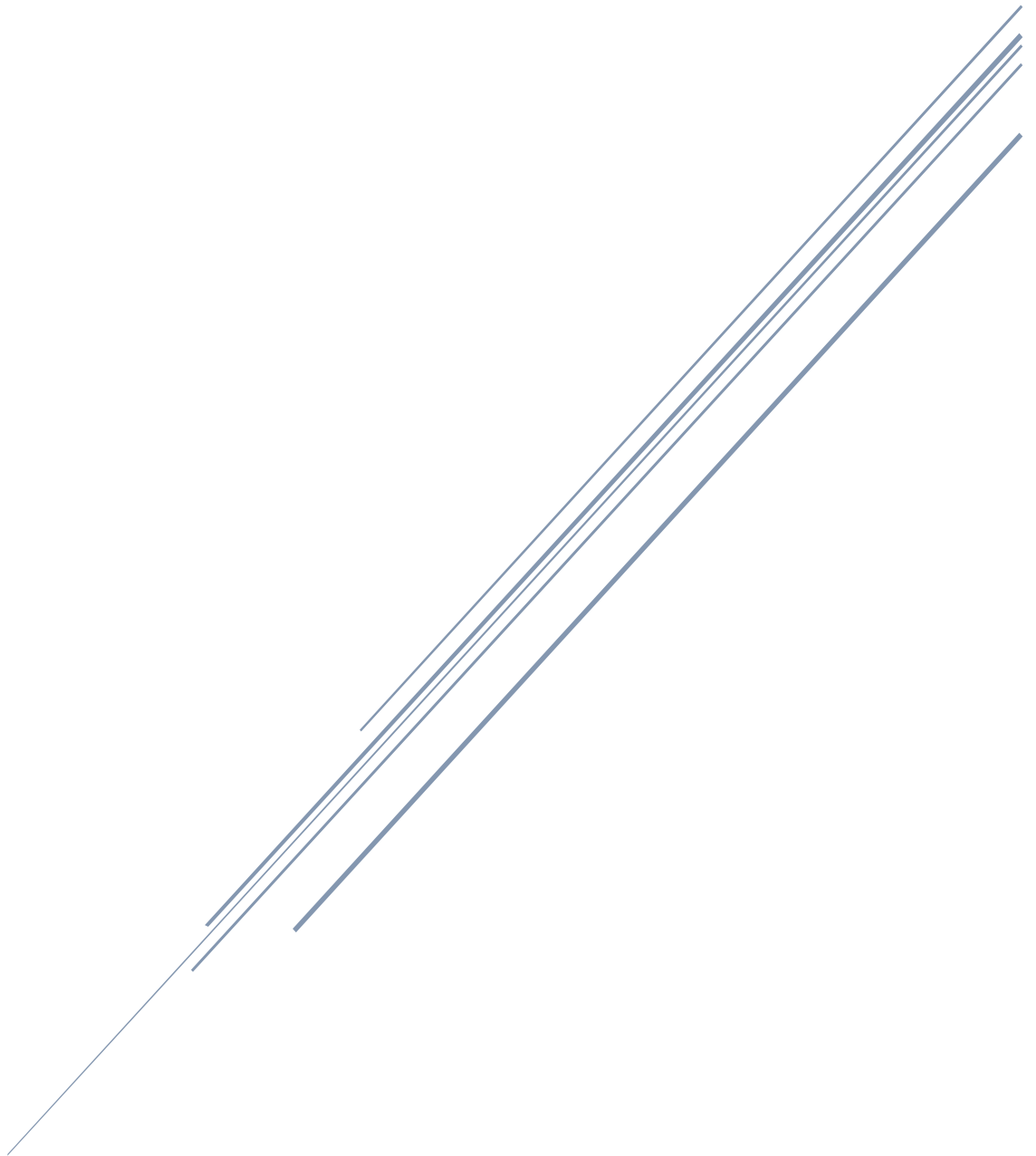


# *BASE DE DATOS FARMACIA*

*Documentación técnica*



*Curso SQL*  
*José Q. Orozco*

# Índice de contenidos

Consigna.....	2
Modelo de negocio.....	2
Objetivo.....	2
DER Conceptual.....	3
DER Esquemático.....	4
Tablas.....	5
Listado de vistas.....	10
Listado de funciones.....	11
Listado de stored procedures.....	12
Listado de triggers.....	13
Inteligencia empresarial (BI).....	14
Software.....	15

# CURSO SQL #43415

Fecha: 07/07/2023

Entrega: Proyecto final

Alumno José Q. Orozco

Profesor Cesar L. Aracena

Tutora Vanina Giselle Aranda

## Se debe entregar

- Descripción de la temática de la base de datos
  - Diagramas de entidad relación de la base de datos
  - Listado de las tablas que comprenden la base de datos, con descripción de cada tabla, listado de campos, abreviaturas de nombres de campos, nombres completos de campos, tipos de datos, tipo de clave (foránea, primaria, índice(s))
  - Listado de Vistas más una descripción detallada, su objetivo, y qué tablas las componen.
  - Listado de Funciones que incluyan una descripción detallada, el objetivo para la cual fueron creadas y qué datos o tablas manipulan y/o son implementadas.
  - Listado de Stored Procedures con una descripción detallada, qué objetivo o beneficio aportan al proyecto, y las tablas que lo componen y/o tablas con las que interactúa.
  - Listado de Triggers con una descripción detallada, el objetivo para la cual fueron creadas y qué datos o tablas manipulan y/o son implementadas.
- 

## El modelo de negocios será el de una cadena de farmacias con varias sucursales:

Esta temática implica la creación de una base de datos para gestionar la venta de productos y servicios de una farmacia, lo que incluye información sobre medicamentos, productos de belleza, servicios, precios, clientes, empleados, proveedores, niveles de stock, entre otros. Las compras se pueden realizar tanto presencial en cualquiera de las sucursales, como vía web.

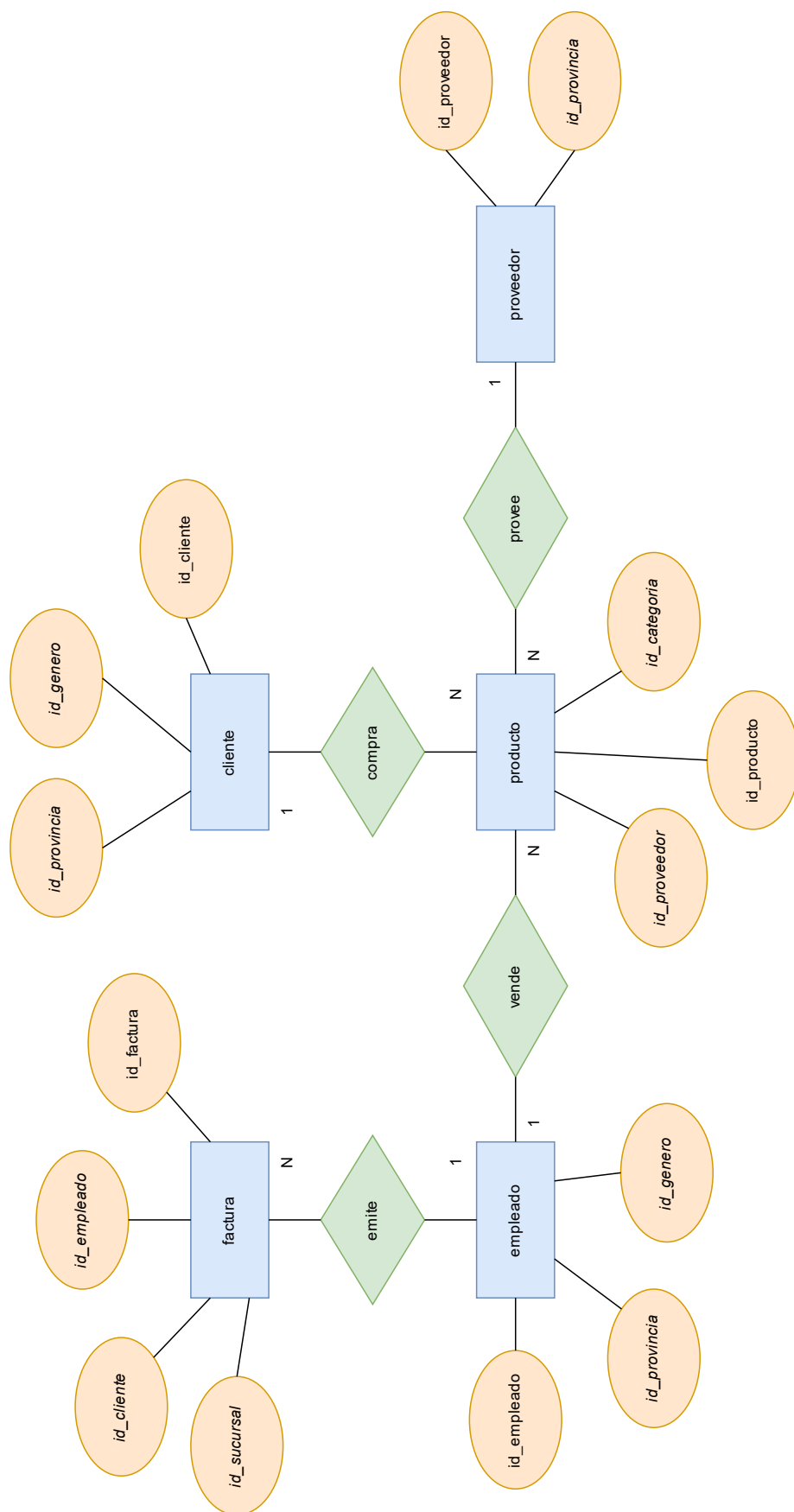
## Objetivo

Optimizar las operaciones, mejorar la atención al cliente, tomar decisiones basadas en datos y garantizar el cumplimiento normativo. La base de datos permite gestionar eficientemente el inventario, ofrecer atención personalizada a los clientes, realizar análisis de datos para identificar tendencias y patrones, mantener relaciones sólidas con proveedores y cumplir con las regulaciones aplicables.

## Necesidades a cubrir

Satisfacer las necesidades de los clientes en términos de acceso a medicamentos, información y asesoramiento, servicio al cliente de calidad, disponibilidad de productos, cumplimiento normativo y seguridad, y servicios adicionales relacionados con la salud.

# DER Conceptual



# DER Esquemático



## Tablas

<b>Tabla</b>	<b>provincia</b>						
<b>Desc</b>	Contiene nombres de provincias						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_provincia	int		x		AUTO_INCREMENT	id de provincia
	provincia	varchar	50	x			nombre de provincia
<b>Tabla</b>	<b>genero</b>						
<b>Desc</b>	Contiene tipo de genero						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_genero	int		x		AUTO_INCREMENT	id de genero
	genero	varchar	20	x			nombre del genero
<b>Tabla</b>	<b>cliente</b>						
<b>Desc</b>	Contiene informacion de los clientes de la farmacia						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_cliente	int		x		AUTO_INCREMENT	id de cliente
MUL	nombre	varchar	50	x			nombre del cliente
	apellido	varchar	50	x			apellido del cliente
	nacimiento	date		x			nacimiento del cliente
	genero	varchar	20	x			genero del cliente
UNI	email	varchar	100	x	x		email del cliente
	telefono	varchar	20	x			telefono del cliente
	direccion	varchar	300	x			direccion del cliente
	provincia	varchar	50	x			provincia del cliente
<b>Tabla</b>	<b>sucursal</b>						
<b>Desc</b>	Contiene informacion de la sucursal que realizo la venta						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_sucursal	int		x			id de la sucursal
UNI	email	varchar	100	x	x		email de la sucursal
	telefono	varchar	20	x			telefono de la sucursal
	direccion	varchar	300	x			direccion de la sucursal
<b>Tabla</b>	<b>empleado</b>						
<b>Desc</b>	Contiene informacion de los empleados de la farmacia						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_empleado	int		x		AUTO_INCREMENT	id del empleado
MUL	nombre	varchar	50	x			nombre del empleado
	apellido	varchar	50	x			apellido del empleado
	nacimiento	date		x			nacimiento del empleado
	genero	varchar	20	x			genero del empleado
	inicio	date		x			fecha de contratacion
UNI	email	varchar	100	x	x		email del empleado
	telefono	varchar	20	x			telefono del empleado
	direccion	varchar	300	x			direccion del empleado
	provincia	varchar	50	x			provincia del empleado
	activo	tinyint		x		1	empleado activo o no

<b>Tabla</b>	<b>compra_tipo</b>						
<b>Desc</b>	Contiene informacion si la compra fue fisica o via web						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_compra	int		x			id de la compra
	tipo	varchar	20	x			fisica / web
<b>Tabla</b>	<b>pago_tipo</b>						
<b>Desc</b>	Contiene informacion si la compra se hizo en efectivo, debito o con tarjeta de credito						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_pago	int		x			id del pago
	tipo	varchar	20	x			tarjeta / efectivo / etc
<b>Tabla</b>	<b>factura</b>						
<b>Desc</b>	Contiene informacion de la factura entregada al cliente						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_factura	int		x		AUTO_INCREMENT	id de la factura
MUL	fecha	date		x		CURRENT_DATE	fecha de facturacion
	hora	time		x		CURRENT_TIME	hora de facturacion
	precio_bruto	dec	11,2	x		0	precio bruto
	impuesto	dec	5,2	x		0	impuesto
	total	dec	11,2	x		0	bruto mas impuestos
FK	id_empleado	int		x			id del empleado vendedor
FK	id_cliente	int		x			id del cliente
FK	id_pago	int		x			id del tipo de pago
FK	id_compra	int		x			id del tipo de compra
FK	id_sucursal	int		x			id de la sucursal
<b>Tabla</b>	<b>categoria_producto</b>						
<b>Desc</b>	Contiene informacion de la categoria del producto						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_categoria	int		x			id de la categoria
	tipo	varchar	20	x			medicamentos, servicios, etc
<b>Tabla</b>	<b>proveedor</b>						
<b>Desc</b>	Contiene informacion de los proveedores						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_proveedor	int		x		AUTO_INCREMENT	id del proveedor
MUL	nombre	varchar	50	x			nombre del proveedor
	apellido	varchar	50	x			apellido del proveedor
	email	varchar	100	x	x		email del proveedor
	telefono	varchar	20	x			telefono del proveedor
	direccion	varchar	300	x			direccion del proveedor
	provincia	varchar	50	x			provincia del proveedor

<b>Tabla</b>	<b>producto</b>						
<b>Desc</b>	Contiene informacion de los productos en venta						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_producto	int		x		AUTO_INCREMENT	id del producto
	nombre	varchar	100	x			nombre del producto
	precio	dec	11,2	x			precio del producto
MUL	farmaco	varchar	100			NULL	farmaco del producto, si tiene
FK	id_categoria	int		x			categoria de producto
FK	id_proveedor	int					id del proveedor
	activo	tinyint		x		1	producto activo o no
<b>Tabla</b>	<b>detalle_venta</b>						
<b>Desc</b>	Contiene informacion detallada de la venta						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_venta	int		x		AUTO_INCREMENT	id de venta
FK	id_factura	int		x			id de la factura
FK	id_producto	int		x			id del producto
	cantidad	int		x		1	cantidad de producto
	precio	dec	11,2	x		0	precio del producto
	subtotal	dec	11,2				precio de producto * cantidad
<b>Tabla</b>	<b>stock</b>						
<b>Desc</b>	Contiene informacion del stock de productos						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_stock	int		x		AUTO_INCREMENT	id de stock
FK	id_producto	int		x			id del producto
	cantidad	int		x		0	cantidad del producto
	stock_min	int		x		0	cantidad minima
	stock_max	int		x		0	cantidad maxima
<b>Tabla</b>	<b>salario</b>						
<b>Desc</b>	Contiene registro sobre pago de salario de empleados						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_salario	int		x		AUTO_INCREMENT	id del salario
FK	id_empleado	int		x			id del empleado
	fecha	date		x			fecha de pago
	monto	dec	11,2	x		0	monto del pago
<b>Tabla</b>	<b>lote</b>						
<b>Desc</b>	Contiene informacion sobre el vencimiento de productos						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_lote	int		x		AUTO_INCREMENT	id del lote
FK	id_producto	int		x			id del producto
	cantidad	int		x		1	cantidad en el lote
	vencimiento	date					vencimiento del lote



<b>Tabla</b>	<b>registro</b>						
<b>Desc</b>	Registra las entradas y salidas de los empleados						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_registro	int		x		AUTO_INCREMENT	id de registro
FK	id_empleado	int		x			id del empleado
FK	id_sucursal	int		x			id de la sucursal
	entrada	datetime		x			fecha y hora de ingreso
	salida	datetime					fecha y hora de egreso
<b>Tabla</b>	<b>venta_tmp</b>						
<b>Desc</b>	Guarda temporalmente los productos de una venta						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_venta_tmp	int		x		AUTO_INCREMENT	id de venta_tmp
FK	id_empleado	int		x			id del empleado
FK	id_producto	int		x			id del producto
	cantidad	int		x			cantidad de producto
	subtotal	dec	11,2	x			precio de producto * cantidad
	nuevo_stock	int		x			nuevo stock de producto

<b>Tabla</b>	<b>log_nuevo_producto</b>						
<b>Desc</b>	Guarda registro de cuando se inserta un nuevo producto						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_log	int		x		AUTO_INCREMENT	id de log
	id_producto	int		x			id del producto
	fecha	datetime		x			fecha de registro
	usuario	varchar(50)		x			user()
<b>Tabla</b>	<b>log_precios</b>						
<b>Desc</b>	Guarda registro de cambio de precio de los productos						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_log	int		x		AUTO_INCREMENT	id de log
	id_producto	int		x			id del producto
	precio_anterior	dec(11,2)		x			precio anterior
	precio_nuevo	dec(11,2)		x			precio nuevo
	fecha	datetime		x			fecha de registro
	usuario	varchar(50)		x			user()
<b>Tabla</b>	<b>log_cambio_stock</b>						
<b>Desc</b>	Guarda registro de aumento de stock de productos						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_log	int		x		AUTO_INCREMENT	id de log
	id_producto	int		x			id del producto
	fecha	datetime		x			fecha de registro
	usuario	varchar(50)		x			user()
<b>Tabla</b>	<b>log_nueva_factura</b>						
<b>Desc</b>	Guarda registro de cuando se inserta una nueva factura y el monto total						
<b>KEY</b>	<b>COLUMN</b>	<b>TYPE</b>	<b>LENGHT</b>	<b>NOT NULL</b>	<b>UNIQUE</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
PK	id_log	int		x		AUTO_INCREMENT	id de log
	id_factura	int		x			id del producto
	fecha	datetime		x			fecha de registro
	total	dec	11,2	x			total de factura
	usuario	varchar(50)		x			user()

## Listado de Vistas

- **categorias\_productos\_vw**: Vista para conocer todos los productos de la base de datos y sus categorías. Se compone de las tablas **producto** y **categoría**.
- **genero\_cantidad\_vw**: Vista para conocer la cantidad de clientes por género. Se compone de la tabla **género**.
- **lotes\_vencimientos\_vw**: Vista para conocer cuáles lotes de productos vencen dentro de los próximos 15 días o menos, ordenados de forma ascendente. Se compone de las tablas **lote** y **producto**.
- **suelos\_empleados\_vw**: Vista para conocer el sueldo promedio de cada empleado, ordenado de forma descendente. Se compone de las tablas **empleado** y **salario**.
- **ventas\_categoria\_vw**: Vista para conocer la cantidad de ventas por categoría de productos. Se compone de las tablas **venta**, **producto** y **categoría**.
- **ventas\_empleado\_vw**: Vista para conocer la cantidad de ventas de cada empleado, ordenado de forma descendente. Se compone de las tablas **factura** y **empleado**.
- **ventas\_provincias\_vw**: Vista para conocer la cantidad de ventas por provincia ordenados alfabéticamente. Se compone de las tablas **factura** y **provincia**.
- **facturación\_total\_vw**: Vista para devolver el total de ventas por año ordenados de forma ascendente.
- **proveedor\_total\_vw**: Vista para conocer la cantidad de productos que provee cada proveedor ordenado por cantidad de forma descendente. Se compone de las tablas **producto** y **proveedor**.
- **detalle\_facturacion\_vw**: Vista para devolver el detalle de facturación, ordenado por fecha de forma descendente. Se compone de las tablas **factura**, **empleado**, **cliente**, **pago**, **compra**, **sucursal** y **provincia**.

## Listado de Funciones

- **calcular\_iva\_fn**: Función NO SQL para calcular IVA. Toma como parámetro un valor (tipo dec) (ej: 10.5) y devuelve un valor (tipo dec) (ej: 1.105). Se usa en un stored procedure que genera la factura y calcula el monto total.
- **nivel\_stock\_fn**: Función READS SQL DATA para calcular niveles de stock. Compara la cantidad de un producto contra su stock\_min y stock\_max. Toma como parámetro id\_producto (tipo int) y devuelve el nivel de stock (tipo varchar). Interactúa con la tabla **stock**.
- **salario\_empleado\_fn**: Función READS SQL DATA para conocer el último salario de un empleado. Toma como parámetro id\_empleado (int) y devuelve el monto del último sueldo (tipo dec). Interactúa con la tabla **salario**.
- **total\_factura\_fn**: Función READS SQL DATA para calcular el monto total de una factura (precio bruto \* iva). Toma como parámetro id\_factura (int) y devuelve el monto total (dec). Interactúa con la tabla **factura**.
- **calcular\_edad\_fn**: Función NO SQL para calcular la edad de una persona. Toma como parámetro la fecha de nacimiento (tipo date) y devuelve su edad (tipo int).

## Listado de Stored Procedures

- **registrar\_cliente\_sp**: Procedimiento para registrar un nuevo cliente en la base de datos. Recibe como parámetros nombre, apellido, email, teléfono y dirección (tipo varchar), nacimiento (tipo date), género y provincia (tipo int). Interactúa con la tabla **cliente** y con el trigger **verificar\_edad\_tr**.
- **venta\_x\_categorias\_sp**: Procedimiento para calcular el monto total de ventas por categorías en un rango de fechas especificado. Recibe como parámetros fecha inicial y fecha final (tipo date). Interactúa con las tablas **factura**, **producto** y **categoría**. Devuelve el monto total para cada categoría ordenados de forma descendente.
- **actualizar\_stock\_min\_sp**: Procedimiento para actualizar el stock mínimo de un producto. Recibe como parámetros el id\_producto (tipo int) y el valor de nuevo\_stock\_min (tipo int). Interactúa con la tabla **stock**.
- **actualizar\_stock\_max\_sp**: Procedimiento para actualizar el stock máximo de un producto. Recibe como parámetros el id\_producto (tipo int) y el valor de nuevo\_stock\_max (tipo int). Interactúa con la tabla **stock**.
- **productos\_venta\_tmp\_sp**: Procedimiento para registrar un producto en una tabla buffer (venta\_tmp) que ira guardándolos para luego registrar la venta y la factura. Antes de guardar un producto, comprueba si el mismo tiene stock (devuelve un error si no) o si tiene stock suficiente (devuelve error si no) según la cantidad requerida. Luego, comprueba si el producto no está en la lista (lo agrega si no), y si ya está en la lista actualiza su cantidad. También calcula el subtotal (precio \* cantidad) y calcula el nuevo valor que quedaría en stock. Recibe como parámetros id\_empleado, id\_producto y cantidad (tipo int). Interactúa con la tabla **venta\_tmp**.
- **registrar\_venta\_sp**: Procedimiento para registrar una factura de venta. Recibe como parámetros id\_empleado, id\_cliente, id\_pago, id\_compra, id\_sucursal (tipo int) y el % de impuesto (tipo dec). Primero genera la factura para obtener el id\_factura. Luego según los datos de la tabla buffer y según el id\_empleado, genera el detalle de venta insertando los correspondientes datos en la tabla venta. Después actualiza los datos precio\_bruto, impuesto y total en la factura. Siguiendo paso es actualizar el stock real de los respectivos productos. Por último, vacía la tabla buffer según el id\_empleado. Interactúa con las tablas **producto**, **venta**, **factura**, **venta\_tmp** y **stock**.
- **registrar\_empleado\_sp**: Procedimiento para registrar nuevo empleado. Recibe como parámetros id\_empleado (int), nombre (varchar), apellido (varchar), nacimiento (date), id\_genero (int), inicio (date), email (varchar), teléfono (varchar), dirección (varchar) y id\_provincia (int). Interactúa con la tabla **empleado**.
- **registrar\_proveedor\_sp**: Procedimiento para registrar nuevo proveedor. Recibe como parámetros id\_proveedor (int), nombre (varchar), apellido (varchar), email (varchar), teléfono (varchar), dirección (varchar) y id\_provincia (int). Interactúa con la tabla **proveedor**.
- **registrar\_producto\_sp**: Procedimiento para registrar nuevo producto. Recibe como parámetro nombre (varchar), precio (dec), fármaco (varchar), id\_categoria (int), id\_proveedor (int). Interactúa con la tabla **producto**.
- **actualizar\_producto\_sp**: Procedimiento para actualizar un producto. Recibe como parámetros id\_producto (int), nombre (varchar), precio (dec), fármaco (varchar), id\_categoria (int), id\_proveedor (int), activo (tinyint). Interactúa con la tabla **producto**.

## Listado de Triggers

- **actualizar\_precio\_producto\_tr:** Trigger AFTER UPDATE para registrar la modificación del precio de un producto de la base de datos. Guarda un registro con los datos de `id_producto`, `precio_anterior`, `precio_nuevo`, `fecha` y `usuario`. Interactúa con las tablas **producto** y **log\_precios**.
- **nuevo\_producto\_tr:** Trigger AFTER INSERT para registrar cuando se inserta un nuevo producto en la base de datos. Guarda un registro con los datos de `id_producto`, `fecha` y `usuario`. Interactúa con las tablas **producto** y **log\_nuevo\_producto**.
- **registrar\_facturas\_tr:** Trigger AFTER INSERT para registrar cuando se inserta una nueva factura en la base de datos. Guarda un registro con los datos de `id_factura`, `fecha`, `monto total` y `usuario`. Interactúa con las tablas **factura** y **log\_nueva\_factura**.
- **verificar\_vencimiento\_tr:** Trigger BEFORE INSERT para verificar la fecha de vencimiento del lote antes de su inserción. Si la fecha de vencimiento es menor a la fecha actual, cancela la inserción y devuelve un mensaje de error. Interactúa con la tabla **lote**.
- **aumento\_stock\_tr:** Trigger AFTER UPDATE para registrar cuando se ingresa nuevo stock de un producto. Guarda un registro con los datos de `id_producto`, `fecha` y `usuario`. Solo registra cuando el stock nuevo es mayor al anterior. Interactúa con la tabla **stock** y **log\_cambio\_stock**.
- **verificar\_edad\_tr:** Trigger BEFORE INSERT para verificar la edad de un nuevo cliente. Si el cliente es menor de 18 años, cancela la inserción y devuelve un mensaje de error. Interactúa con la tabla **cliente**.
- **verificar\_empleado\_tr:** Trigger BEFORE INSERT para verificar que un empleado esté activo antes del registro de entrada o salida de una sucursal. Interactúa con la tabla **registro**.

## Gráficos

Basado en las vistas, se puede crear los siguientes gráficos para análisis:



## Software

### Software:

MySQL Community Server 8.0

MySQL Workbench 8.0

Microsoft Word 2016

Microsoft Excel 2016

Microsoft Power BI Desktop

Draw.io 21

Adobe Acrobat 23

### *Datos de conexión a RDS de AWS:*

*host:* database-farmacia.cccupuslvvw9.us-east-2.rds.amazonaws.com

*user:* visitante

*password:* Visitante123