

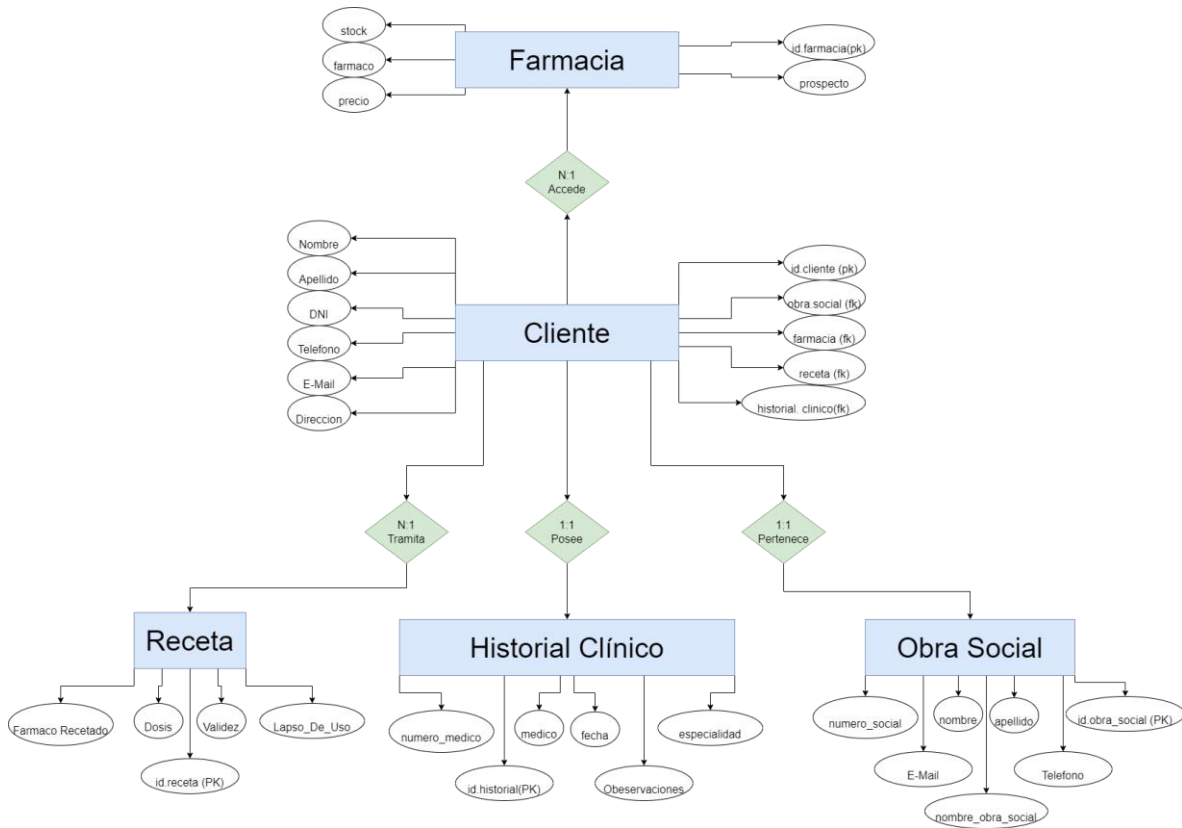
# Introducción y modelo de negocios:

El negocio farmacéutico es algo que ha abundado desde siempre en distintos países del mundo. Este tipo de negocios maneja una gran cantidad de datos que además son de variadas fuentes. Este sistema permite tener el control sobre estos mismos, como son las obras sociales, pacientes, recetas, historiales médicos y en si el stock del mismo local. Este tipo de información a la mano radica en que haya una mayor facilidad a la hora de consultar tanto a quien le estamos dando el fármaco como quien lo cubre.

## Objetivo y situación problemática:

Mi objetivo con este proyecto es crear un sistema organizado y funcional de consultas de cualquier tipo. Donde los datos se puedan relacionar de manera más fácil y se puedan encontrar respuestas mucho más directas a las dudas que el empleado tenga en sí. Si bien las tablas de cada fuente de estos datos es relevante, la busque de combinaciones provoca que se puedan encontrar consultas mas rápidas, sobre si un cliente es jubilado, si su receta es válida, cuantas personas tiene cierta obra social, etc. Además de poder editar, borrar y modificar datos en tiempo real para mantenerlo actualizado. Esto crea un dinamismo a la hora de trabajar dando a los empleados respuestas al instante para satisfacer a sus clientes, sin este tipo de bases de datos, los empleados deberían darse a la tarea de buscar cosa por cosa e ir matcheando que coincide con que, provocando una pérdida de tiempo y dando un margen de error muchísimo más grande. Sin decir que el control de stock sería mucho más fácil de consultar, por supuesto un lugar más eficiente en este ámbito es un lugar preferente para clientes que no quieren perder el tiempo o se quieren sentir seguros con su compra.

# Diagrama Entidad-Relación:



# Listado de tablas:

Tabla Farmacia					
Nombre Columa	Tipo	Descripcion	Clave	Características	
id_farmacia	int	Numero de ID de la farmacia	PK	UNIQUE	NOT NULL
stock_farmaco	varchar(50)	Cantidad disponible del farmaco			NOT NULL
nombre_farmaco	varchar(50)	Nombre del farmaco			NOT NULL
precio_farmaco	float	Precio en stock del farmaco			NOT NULL
descripcion_farmaco	varchar(50)	Descripcion y modo de uso del farmaco			NOT NULL
Tabla Cliente					
Nombre Columa	Tipo	Descripcion	Clave	Características	
nombre_cliente	varchar(50)	Nombre del cliente en cuestion			NOT NULL
apellido_cliente	varchar(50)	Apellido del cliente en cuestion			NOT NULL
dni_cliente	int	DNI del cliente en cuestion		UNIQUE	NOT NULL
telefono_cliente	int	Telefono de contacto del cliente			NULL
email_cliente	varchar(50)	E-Mail de contacto del cliente		UNIQUE	NULL
direccion_cliente	varchar(50)	Direccion donde reside el cliente			NOT NULL
id_cliente	int	ID del cliente en cuestion	PK	UNIQUE	NOT NULL
id_obrasocial	int	ID de la obra social del cliente	FK		NOT NULL
id_farmacia	int	ID de la farmacia donde va el cliente	FK		NOT NULL
num_receta	int	Numero de referencia de la receta del farmaco	FK		NOT NULL
id_historialclinico	int	ID del historial clinico del cliente	FK		NOT NULL
Tabla Receta					
Nombre Columa	Tipo	Descripcion	Clave	Características	
id_receta	int	ID de la receta dada	PK	UNIQUE	NOT NULL
dosis	float	Cantidad del farmaco que se deberia usar			NOT NULL
farmaco_recetado	varchar(50)	Nombre del farmaco recetado			NOT NULL
validez_receta	date	Fecha de expiracion de la receta			NOT NULL
frecuencia_uso	float	Frecuencia con la que se deberia de usar el farmaco			NOT NULL
Tabla Historial Clinico					
Nombre Columa	Tipo	Descripcion	Clave	Características	
id_historialclinico	int	ID del historial clinico del paciente	PK	UNIQUE	NOT NULL
nombre_medico	varchar(50)	Nombre del medico frecuentado			NOT NULL
numero_medico	int	Numero del medico frecuentado			NOT NULL
observacion	varchar(50)	Observaciones extras			NULL
especialidad	varchar(50)	Especialidad del medico			NOT NULL
fecha_citada	date	Fecha que fue citado antes de la ultima receta			NOT NULL
Tabla Obra Social					
Nombre Columa	Tipo	Descripcion	Clave	Características	
id_obrasocial	int	ID de la obra social	PK	UNIQUE	NOT NULL
telefono_cliente	int	Telefono de contacto por obra social			NULL
nombre_cliente	varchar(50)	Nombre del asociado			NOT NULL
apellido_cliente	varchar(50)	Apellido del asociado			NOT NULL
nombre_obrasocial	varchar(50)	Nombre de la obra social			NOT NULL
email_cliente	varchar(50)	E-Mail del asociado			NULL
numero_socio	int	Numero de indentificacion del asociado		UNIQUE	NOT NULL

Siendo en este caso particular la tabla CLIENTE la tabla principal que recibe las FK. Los datos hardcoded están divididos y testeados en distintos archivos .csv (Lee anexo “Explicación de importación de datos.pdf”).

# Tablas:

Las tablas que contiene esta base de datos son 5:

- 1- v\_cliente: El objetivo de esta tabla es ver un resumen general de los clientes que acceden a la farmacia, a diferencia de la tabla original de cliente se evitan datos que puedan llegar a confundir como las FK, estos datos son sacados y lo componen la misma tabla CLIENTE.
- 2- v\_farmaco: El objetivo de esta tabla es ver un resumen de los fármacos que están listados en la farmacia de manera resumida, evitando el id de la farmacia y el stock, esto es útil a la hora de ver en llegada de nuevos fármacos, cuales ya están registrados en el sistema. Estos datos vienen de la misma tabla FARMACIA.
- 3- v\_obra\_cliente: Este listado es útil a la hora de cobrar, ya que muestra a que obra social y que datos de la misma pertenece cada cliente, perfecto para no perder el rumbo del listado de Clientes y poder progresar en su historial clínico o tener una idea de que cobrar dependiendo de donde estén asegurados, esta tabla en particular contiene datos de la tabla CLIENTE y OBRA SOCIAL.
- 4- v\_historial\_cliente: Este listado es útil para conocer el historial clínico de los clientes de forma limpia, mostrando directamente, junto a los datos del cliente, su médico, historial, observación y numero de contacto. Perfecto para consultas de último momento, esta tabla toma datos de CLIENTE e HISTORIAL CLINICO.
- 5- v\_receta\_cliente: Este listado es útil para el ingreso de recetas especificando el cliente al que se le da, no tanto para consulta sino a modo de historial de recetas. Los datos donde se toma este listado son de CLIENTE y RECETA

# Funciones:

Esta base de datos contiene dos funciones en particular:

- 1- farmaco\_en\_stock: Esta función crea un listado de los fármacos mostrando cuales están en stock y cuáles no, en un listado aparte, es literalmente una consulta de stock donde recibe dos datos directo de FARMACIA; el nombre del fármaco y el stock (Que tiene los valores 0 y 1), eventualmente si recibe 0, es que no hay y si recibe 1, hay disponible. Esta función devuelve un varchar "No hay" y "Si Hay".
- 2- dato\_cliente\_general: Esta es una función de consulta también que utiliza el dato dni\_cliente de CLIENTE, este listado lo que provoca es ver de manera dinámica que clientes tienen más de 60 años, la utilidad de esto es para dar descuento de jubilados o saber que medicamento es apto para el mismo.

## Procedures:

Esta base de datos contiene dos stored procedures;

- 1- orden\_farmaco: Este procedimiento lo que provoca es un orden de los fármacos por precio, del menor a mayor, esto sirve para tener una idea de que ofertar a un cliente que su presupuesto no se ajusta al precio del posible fármaco ideal. Este toma el dato de Farmacia; precio\_farmaco y los ordena de manera descendiente.
- 2- ingreso\_farmaco: Este procedure lo que provoca es ingresar nuevos fármacos al listado, ingresando en el listado FARMACIA los valores de nombre, precio, descripción, un ID por defecto y stock por defecto de un fármaco nuevo.

## Trigger:

Esta base de datos contiene dos triggers:

- 1- agregar\_farmaco: Este trigger se ejecuta cuando se interactúa con la tabla FARMACIA para editar los fármacos dentro, provocando un log dependiendo la situación, 'insert' a la hora de ingresar un nuevo fármaco, mostrando quien lo hizo, a qué hora, que día y los datos ingresados, 'update', cuando se edita un fármaco, mostrando los datos anteriores (Update) y los datos antiguos antes de su cambio (Pre-Update) y 'delete' que solo hace un display de los datos que se borran y el registro de cuándo y quien lo hizo.
- 2- receta\_log: Este trigger se ejecuta cuando se interactúa con la tabla RECETA para editar las recetas dentro, provocando un log dependiendo la situación, 'insert' a la hora de ingresar la nueva receta, mostrando quien lo hizo, a qué hora, que día y los datos ingresados, 'update', cuando se edita una receta, mostrando los datos anteriores (Update) y los datos antiguos antes de su cambio (Pre-Update) y 'delete' que solo hace un display de los datos que se borran y el registro de cuándo y quien lo hizo.

## Creacion de usuarios:

Este proyecto contiene la creación de dos usuarios, sujetoLectura que crea un usuario que radica en el uso de contraseña para ingresar, este permite nada mas consultar los datos de las tablas a través de solo lectura y luego sujetoImportante, siendo este el encargado de poder editar, ingresar, actualizar o borrar datos de dichas tablas.

Como punto y aparte, este sistema también contiene rollbacks y savepoints a modo de prueba, que pueden ser utilizados en el futuro como alternativa al backup de los datos. Todo esto creado

por Workbench, este proyecto sigue en proceso y se puede pulir aun mas, pero este es a gran escala la idea del mismo.