

Base de Datos

Unidad III

PROYECTO

Integrantes:

* 21170139: Juan Ángel Castañeda Chávez.
* 21170073: Victor Emmanuel Cabello Rodriguez.
* 21170092: Manuel David Hernández Morales.
* 21170129: Gema Daniela Castañeda Barraza.
* 21170180: Dylan González Flores.

**Cuatrimestre:** 3 | **Sección:** “C” **Torreón Coahuila 08/Agosto/2022**



Tabla de contenido

[PLANTEAMIENTO 4](#_Toc110940866)

[Modelo de negocio. 4](#_Toc110940867)

[Procesos 5](#_Toc110940868)

[Objetivos. 5](#_Toc110940869)

[DISEÑO 6](#_Toc110940870)

[Modelo de Base de datos 6](#_Toc110940871)

[Esquema 7](#_Toc110940872)

[SCRIPT 8](#_Toc110940873)

[TABLAS 13](#_Toc110940874)

[DICCIONARIO DE DATOS 17](#_Toc110940875)

[GESTION 19](#_Toc110940876)

[Administrador 19](#_Toc110940877)

[Doctor 19](#_Toc110940878)

[Cliente 19](#_Toc110940879)

[Sentencias 20](#_Toc110940880)

[Administrador 20](#_Toc110940881)

[Doctor 25](#_Toc110940882)

[Cliente 25](#_Toc110940883)

# PLANTEAMIENTO

## Modelo de negocio.

La farmacia Santa Teresita inicio en diciembre de 1990 en el Municipio de Tlahualilo, Durango. Sus actividades están dedicada a la venta de estupefacientes, alcaloides, barbitúricos, oxitócicos, corticoides y sicofármacos.

El modelo de negocio de la farmacia se basa en la venta directa de medicamentos al público, así como realizar citas mediante la persona encargada de atender el negocio.

La farmacia “Santa Teresita” cuenta con unas personas en el local de ventas, es la dueña del negocio;

Las citas que realizan los pacientes se controlan por turnos, estos se obtienen en con la persona encargada de atender el negocio. Estos turnos se encuentran enumerados, y conforme llega la gente a pedirlo, es el orden en el que se va a atender. Se cuenta con tres doctores para atender a los pacientes .

Los medios de pago que se ofrecerán son contado-efectivo, tarjetas de débito y crédito

La farmacia brinda un servicio de turno 12 horas diarias de lunes a viernes y de 8 horas los días sábados y domingos.

El mercado en el que está dirigido el negocio está compuesto por una población de características urbana con una gran dispersión geográfica. Si bien el punto en el cual se encuentra establecido el local de negocio es el centro urbano de la localidad de Tlahualilo, la densidad poblacional del mismo es muy buena

## Procesos

* **Ventas**

Para el proceso de ventas se tomo en cuenta la alta demanda que hay en el negocio de la venta de medicamentos, ya que estas ventas son administradas con actividades tradicionales y a largo plazo generara grandes pérdidas de tiempo y de dinero, desconocimiento del stock de los productos en el almacén, perdidas de mercancía, falta de control sobre las ventas y compras, dificultad al tomar decisiones, pérdida de tiempo para los clientes y falta de competitividad con las farmacias vecinas, esto ocasiona una reducción de clientes y la disminución de las ventas.

* **Citas.**

El proceso de citas se tomo en cuenta, ya que el proceso mediante el cual se registran es muy obsoleto y no se lleva un control, departe del que atiende el negocio, y ni por los médicos. Esto facilitara tanto a los tres actores (administrador, doctor cliente) a llevar un orden

## Objetivos.

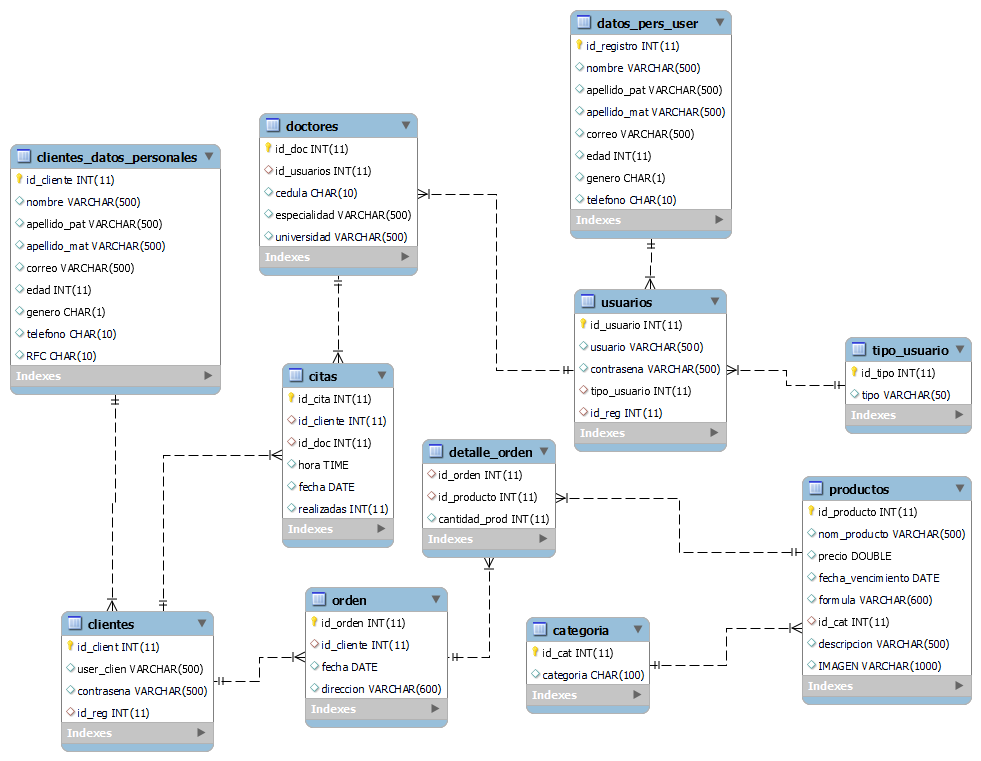
Los objetivos que se pretende con esta aplicación web se dividen en dos los son

* **General:** Diseñar una aplicación web que permita la venta de medicamentos y generación de citas en la Farmacia Santa Teresita.
* **Específicas:**
* Gestionar las **ventas** que se realicen en la aplicación así como la creación de reportes.
* Facilitar la revisión y control para las **citas**, para beneficio del doctor así como del cliente.

# 

# DISEÑO

## Modelo de Base de datos



## Esquema

**datos\_pers\_user** (registro, nombre, apellido\_pat, apellido\_mat, correo, edad, genero, teléfono)

**usuarios** (id usuario, usuario, contraseña, tipo\_usuario, id\_reg)

**tipo\_usuario** (id\_tipo, tipo)

**Doctores** (id\_doc, id\_usuarios, cedula, especialidad, universidad)

**Citas** (id\_cita, id\_cliente, id\_doc, hora, fecha, realizadas)

**Clientes** (id\_client, id\_reg, user\_clien, contraseña)

**clientes\_datos\_personales** (id\_orden, nombre, apellido\_pat, apellido\_mat, correo, edad, genero, teléfono, RFC)

**Orden** (id\_orden, id\_cliente, fecha, dirección)

**detalle\_orden** (id\_orden, id\_producto, cantidad\_prod)

**productos** (id\_producto, nom\_producto, precio, fecha\_vencimiento, formula, id\_cat, descripción, IMAGEN)

**categoría** (id\_cat, categoría)

# SCRIPT

-- MySQL Script generated by MySQL Workbench

-- Tue Aug 9 12:10:21 2022

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema proyecto

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema proyecto

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `proyecto` DEFAULT CHARACTER SET utf8mb4 ;

USE `proyecto` ;

-- -----------------------------------------------------

-- Table `proyecto`.`categoria`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `proyecto`.`categoria` (

`id\_cat` INT(11) NOT NULL AUTO\_INCREMENT,

`categoria` CHAR(100) NULL DEFAULT NULL,

PRIMARY KEY (`id\_cat`))

ENGINE = InnoDB

AUTO\_INCREMENT = 11

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `proyecto`.`clientes\_datos\_personales`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `proyecto`.`clientes\_datos\_personales` (

`id\_cliente` INT(11) NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(500) NULL DEFAULT NULL,

`apellido\_pat` VARCHAR(500) NULL DEFAULT NULL,

`apellido\_mat` VARCHAR(500) NULL DEFAULT NULL,

`correo` VARCHAR(500) NULL DEFAULT NULL,

`edad` INT(11) NULL DEFAULT NULL,

`genero` CHAR(1) NULL DEFAULT NULL,

`telefono` CHAR(10) NULL DEFAULT NULL,

`RFC` CHAR(10) NULL DEFAULT NULL,

PRIMARY KEY (`id\_cliente`))

ENGINE = InnoDB

AUTO\_INCREMENT = 6

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `proyecto`.`clientes`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `proyecto`.`clientes` (

`id\_client` INT(11) NOT NULL AUTO\_INCREMENT,

`user\_clien` VARCHAR(500) NULL DEFAULT NULL,

`contrasena` VARCHAR(500) NULL DEFAULT NULL,

`id\_reg` INT(11) NULL DEFAULT NULL,

PRIMARY KEY (`id\_client`),

INDEX `fk\_usclien` (`id\_reg` ASC) VISIBLE,

CONSTRAINT `fk\_usclien`

FOREIGN KEY (`id\_reg`)

REFERENCES `proyecto`.`clientes\_datos\_personales` (`id\_cliente`))

ENGINE = InnoDB

AUTO\_INCREMENT = 6

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `proyecto`.`datos\_pers\_user`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `proyecto`.`datos\_pers\_user` (

`id\_registro` INT(11) NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(500) NULL DEFAULT NULL,

`apellido\_pat` VARCHAR(500) NULL DEFAULT NULL,

`apellido\_mat` VARCHAR(500) NULL DEFAULT NULL,

`correo` VARCHAR(500) NULL DEFAULT NULL,

`edad` INT(11) NULL DEFAULT NULL,

`genero` CHAR(1) NULL DEFAULT NULL,

`telefono` CHAR(10) NULL DEFAULT NULL,

PRIMARY KEY (`id\_registro`))

ENGINE = InnoDB

AUTO\_INCREMENT = 6

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `proyecto`.`tipo\_usuario`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `proyecto`.`tipo\_usuario` (

`id\_tipo` INT(11) NOT NULL AUTO\_INCREMENT,

`tipo` VARCHAR(50) NULL DEFAULT NULL,

PRIMARY KEY (`id\_tipo`))

ENGINE = InnoDB

AUTO\_INCREMENT = 3

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `proyecto`.`usuarios`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `proyecto`.`usuarios` (

`id\_usuario` INT(11) NOT NULL AUTO\_INCREMENT,

`usuario` VARCHAR(500) NULL DEFAULT NULL,

`contrasena` VARCHAR(500) NULL DEFAULT NULL,

`tipo\_usuario` INT(11) NULL DEFAULT NULL,

`id\_reg` INT(11) NULL DEFAULT NULL,

PRIMARY KEY (`id\_usuario`),

INDEX `fk\_tipo` (`tipo\_usuario` ASC) VISIBLE,

INDEX `fk\_dtuser` (`id\_reg` ASC) VISIBLE,

CONSTRAINT `fk\_dtuser`

FOREIGN KEY (`id\_reg`)

REFERENCES `proyecto`.`datos\_pers\_user` (`id\_registro`),

CONSTRAINT `fk\_tipo`

FOREIGN KEY (`tipo\_usuario`)

REFERENCES `proyecto`.`tipo\_usuario` (`id\_tipo`))

ENGINE = InnoDB

AUTO\_INCREMENT = 6

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `proyecto`.`doctores`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `proyecto`.`doctores` (

`id\_doc` INT(11) NOT NULL AUTO\_INCREMENT,

`id\_usuarios` INT(11) NULL DEFAULT NULL,

`cedula` CHAR(10) NULL DEFAULT NULL,

`especialidad` VARCHAR(500) NULL DEFAULT NULL,

`universidad` VARCHAR(500) NULL DEFAULT NULL,

PRIMARY KEY (`id\_doc`),

INDEX `fk\_docuser` (`id\_usuarios` ASC) VISIBLE,

CONSTRAINT `fk\_docuser`

FOREIGN KEY (`id\_usuarios`)

REFERENCES `proyecto`.`usuarios` (`id\_usuario`))

ENGINE = InnoDB

AUTO\_INCREMENT = 4

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `proyecto`.`citas`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `proyecto`.`citas` (

`id\_cita` INT(11) NOT NULL AUTO\_INCREMENT,

`id\_cliente` INT(11) NULL DEFAULT NULL,

`id\_doc` INT(11) NULL DEFAULT NULL,

`hora` TIME NULL DEFAULT NULL,

`fecha` DATE NULL DEFAULT NULL,

`realizadas` INT(11) NULL DEFAULT NULL,

PRIMARY KEY (`id\_cita`),

INDEX `fk\_cli` (`id\_cliente` ASC) VISIBLE,

INDEX `fk\_doct` (`id\_doc` ASC) VISIBLE,

CONSTRAINT `fk\_cli`

FOREIGN KEY (`id\_cliente`)

REFERENCES `proyecto`.`clientes` (`id\_client`),

CONSTRAINT `fk\_doct`

FOREIGN KEY (`id\_doc`)

REFERENCES `proyecto`.`doctores` (`id\_doc`))

ENGINE = InnoDB

AUTO\_INCREMENT = 3

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `proyecto`.`orden`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `proyecto`.`orden` (

`id\_orden` INT(11) NOT NULL AUTO\_INCREMENT,

`id\_cliente` INT(11) NULL DEFAULT NULL,

`fecha` DATE NULL DEFAULT NULL,

`direccion` VARCHAR(600) NULL DEFAULT NULL,

PRIMARY KEY (`id\_orden`),

INDEX `fk\_cliente` (`id\_cliente` ASC) VISIBLE,

CONSTRAINT `fk\_cliente`

FOREIGN KEY (`id\_cliente`)

REFERENCES `proyecto`.`clientes` (`id\_client`))

ENGINE = InnoDB

AUTO\_INCREMENT = 4

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `proyecto`.`productos`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `proyecto`.`productos` (

`id\_producto` INT(11) NOT NULL AUTO\_INCREMENT,

`nom\_producto` VARCHAR(500) NULL DEFAULT NULL,

`precio` DOUBLE NULL DEFAULT NULL,

`fecha\_vencimiento` DATE NULL DEFAULT NULL,

`formula` VARCHAR(600) NULL DEFAULT NULL,

`id\_cat` INT(11) NULL DEFAULT NULL,

`descripcion` VARCHAR(500) NULL DEFAULT NULL,

`IMAGEN` VARCHAR(1000) NULL DEFAULT NULL,

PRIMARY KEY (`id\_producto`),

INDEX `fk\_cat` (`id\_cat` ASC) VISIBLE,

CONSTRAINT `fk\_cat`

FOREIGN KEY (`id\_cat`)

REFERENCES `proyecto`.`categoria` (`id\_cat`))

ENGINE = InnoDB

AUTO\_INCREMENT = 4

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `proyecto`.`detalle\_orden`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `proyecto`.`detalle\_orden` (

`id\_orden` INT(11) NULL DEFAULT NULL,

`id\_producto` INT(11) NULL DEFAULT NULL,

`cantidad\_prod` INT(11) NULL DEFAULT NULL,

INDEX `fk\_orden` (`id\_orden` ASC) VISIBLE,

INDEX `fk\_prod` (`id\_producto` ASC) VISIBLE,

CONSTRAINT `fk\_orden`

FOREIGN KEY (`id\_orden`)

REFERENCES `proyecto`.`orden` (`id\_orden`),

CONSTRAINT `fk\_prod`

FOREIGN KEY (`id\_producto`)

REFERENCES `proyecto`.`productos` (`id\_producto`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

# TABLAS

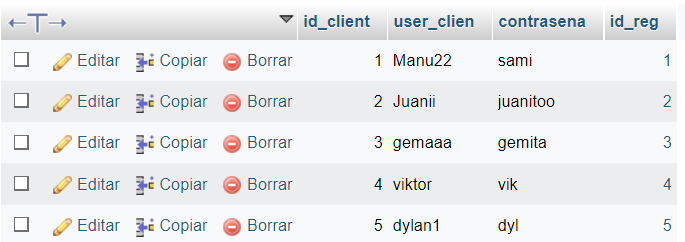
1. Categoria



1. Citas



1. Clientes



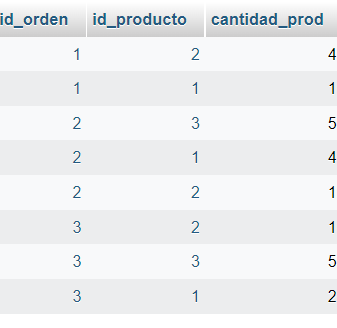
1. Clientes\_datos\_personales.



1. Datos\_pers\_user



1. Detalle\_orden



1. Doctores



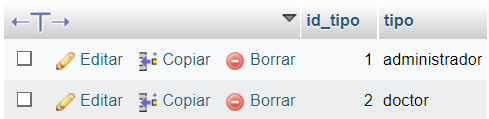
1. Orden



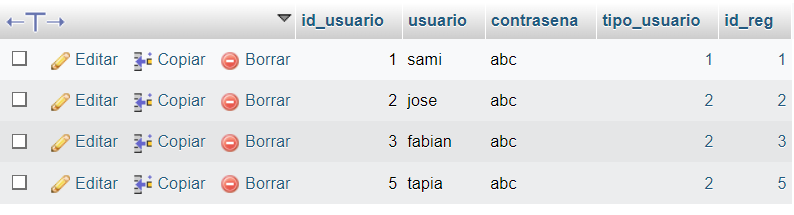
1. Productos



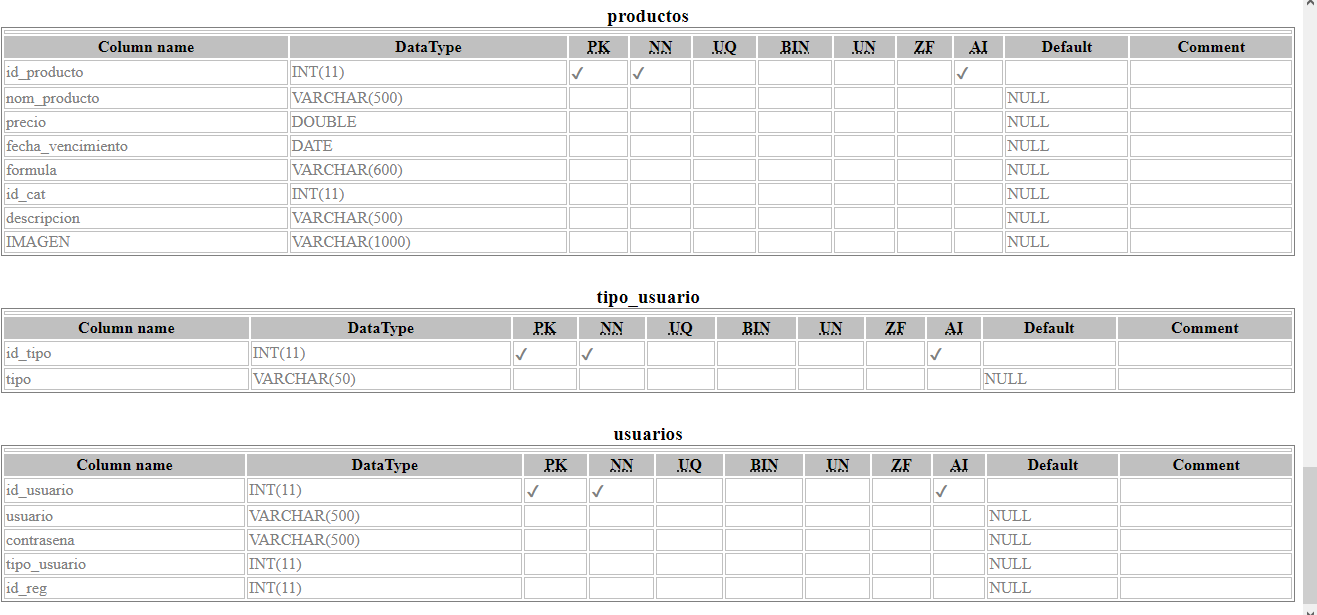
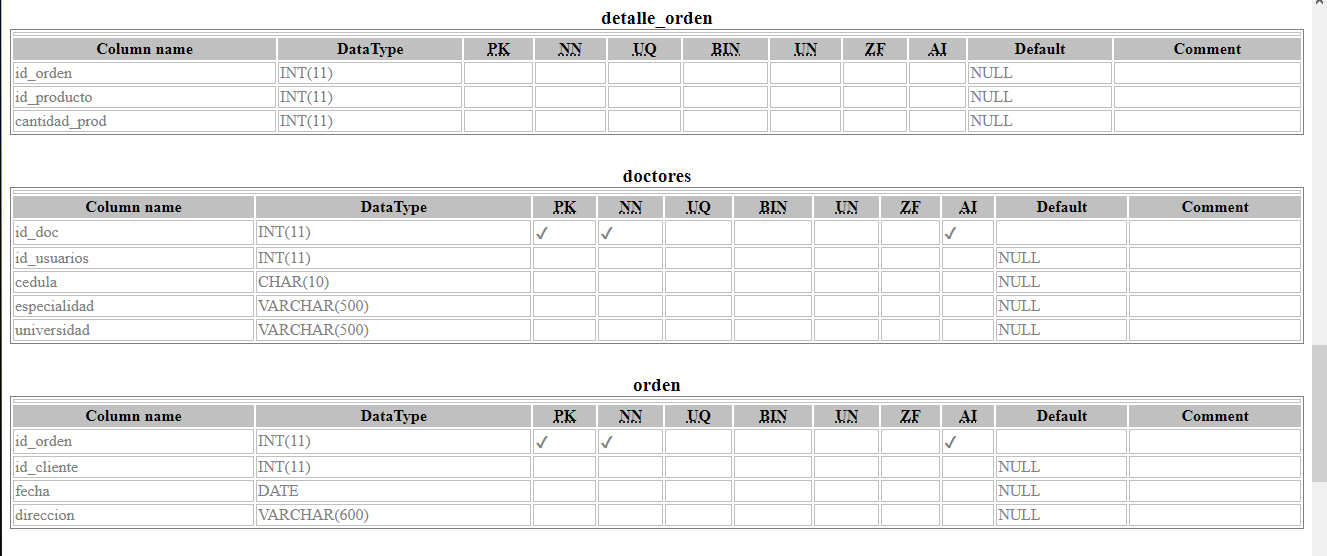
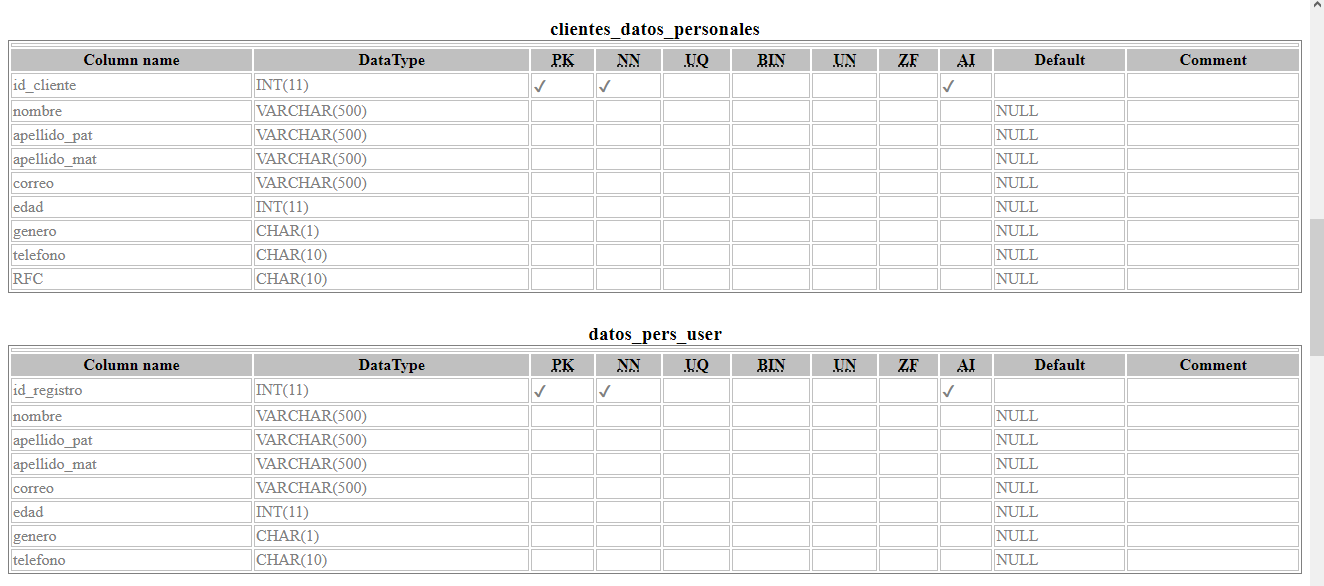
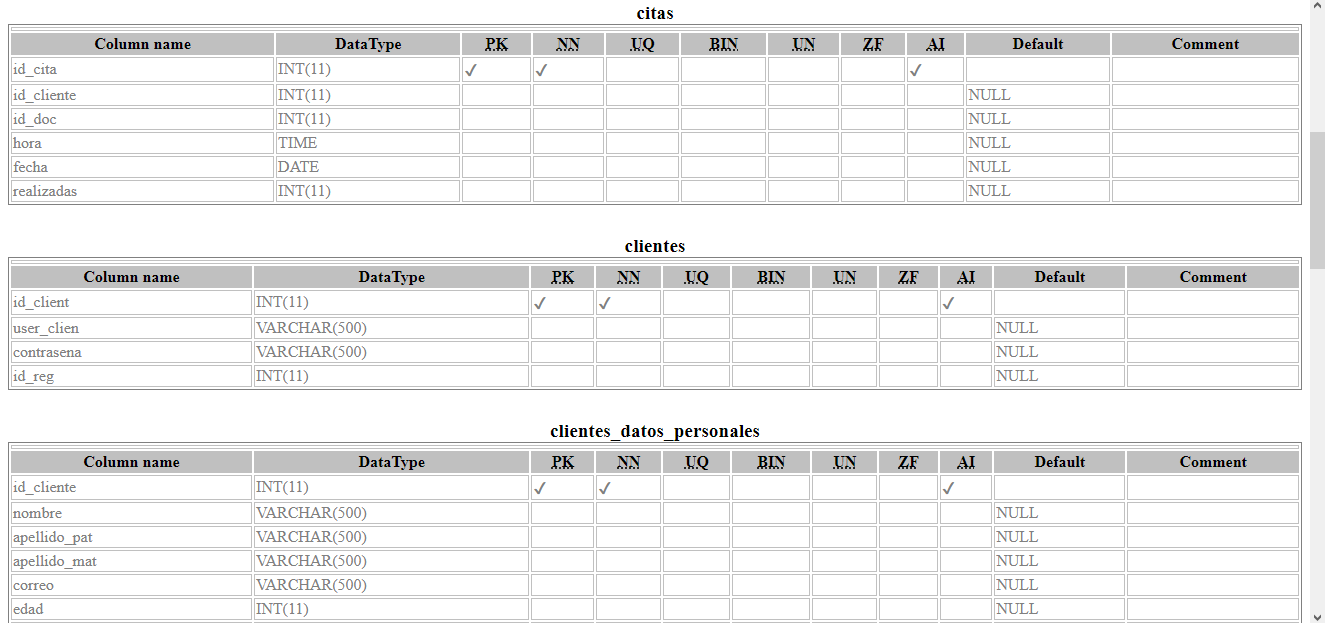
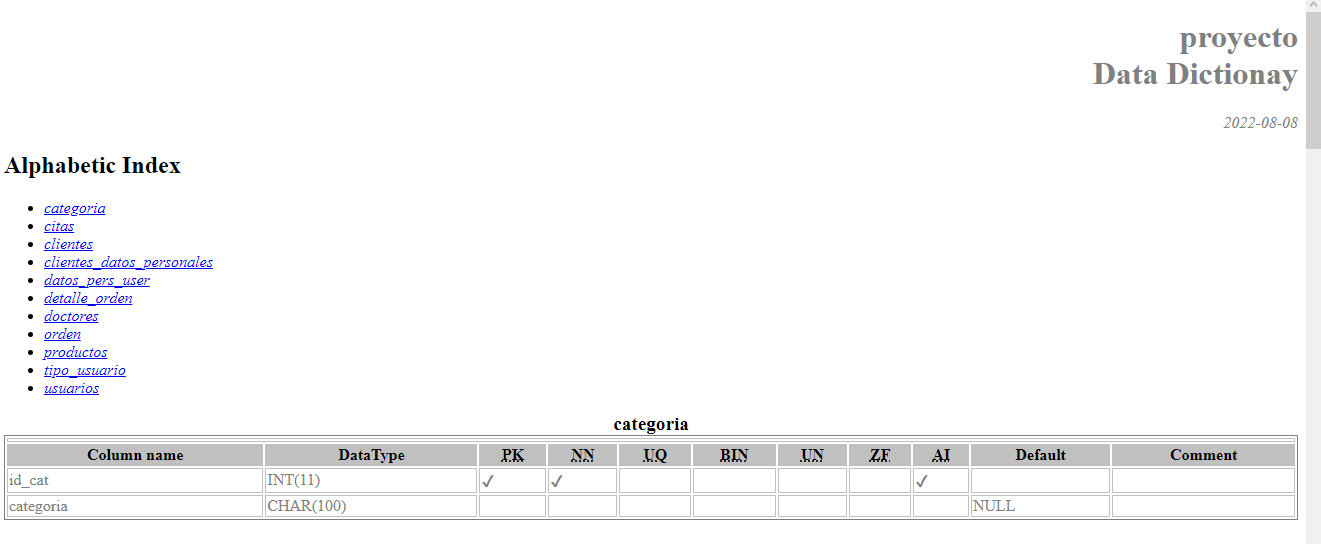
1. Tipo\_usuario



1. Usuarios



# DICCIONARIO DE DATOS



# GESTION

* El proyecto es gestionado por tres roles, los cuales realizar las actividades que les corresponde a cada uno atreves de la aplicación web. Los tres roles son.

## Administrador

El administrador será el encargado de administrar la página web, este realizara las siguientes actividades:

* Registrar productos en el sistema.
* Eliminar productos
* Realizar reportes del día.
* Ver las citas generadas por los clientes.
* Ver las compras por cliente.

## Doctor

El rol del doctor en la aplicación web, tendrá las actividades de gestión de citas, actividades a realizar, serán:

* Ver las citas que tendrá durante el día.
* Ver las citas anteriores.
* Gestionar las citas que ya se han realizado, y las que no se han realizado.

## Cliente

El cliente en la página web, será el actor principal, ya que con las actividades que realice le dará funcionalidades a la página. Las acciones que podrá realizar serán:

* Loguear
* Ver productos.
* Comprar Productos.
* Realizar la cita
* Escoger el medico.
* Asignar hora y fecha de la cita.
* Ver su carrito de compras

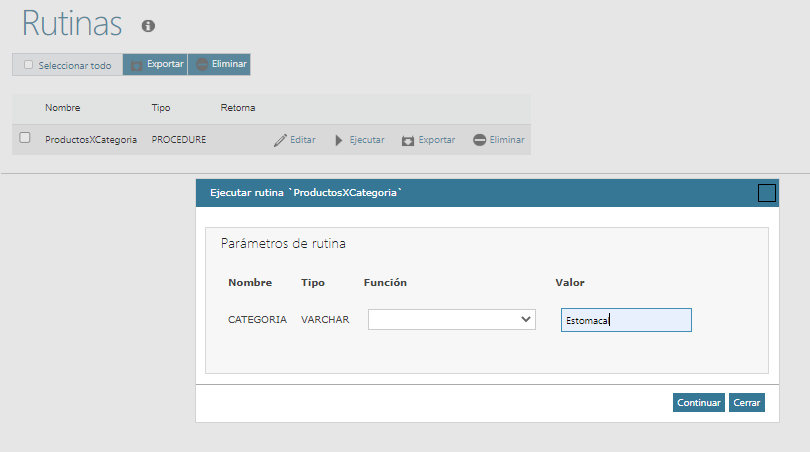
## Sentencias

Las siguientes sentencias que se mostraran en este apartado, son las que se utilizaran para la aplicación web. Estas están en orden de cada rol.

### *Administrador*

1. Ver los productos existentes por determinada categoría (Procedimiento almacenado)

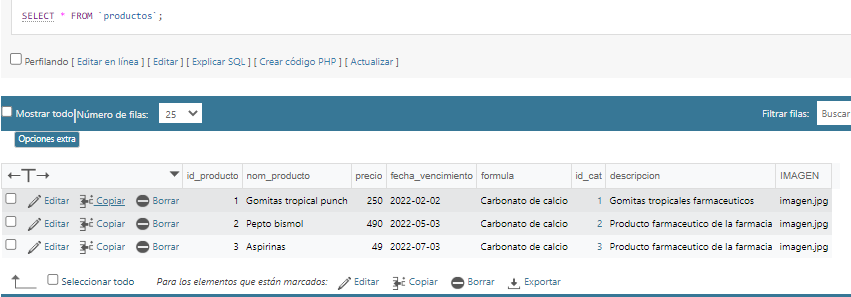
create procedure ProductosXCategoria (IN CATEGORIA varchar (25)) SELECT \* from productos inner join categoria on productos.id\_cat=categoria.id\_cat where categoria.categoria=CATEGORIA;





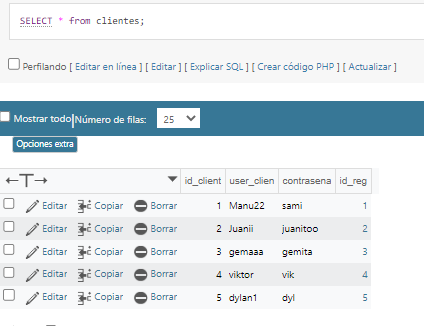
1. Mostrar productos existentes.

**SELECT \* FROM productos**



1. Mostrar datos de ingreso de los clientes registrados.

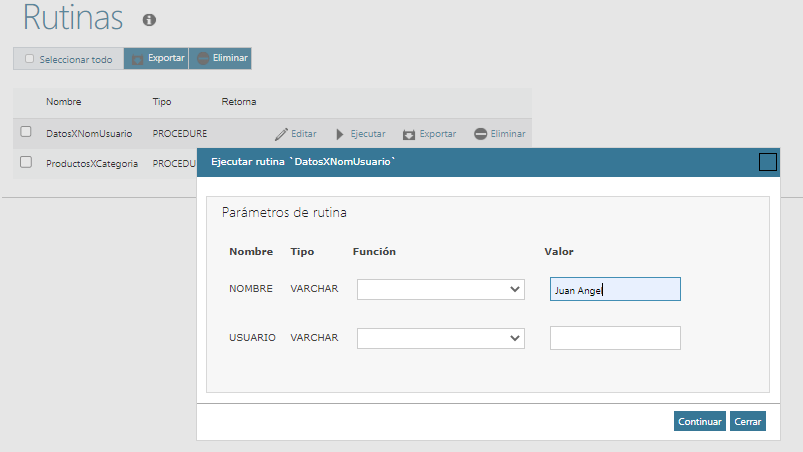
**SELECT \* from clientes**

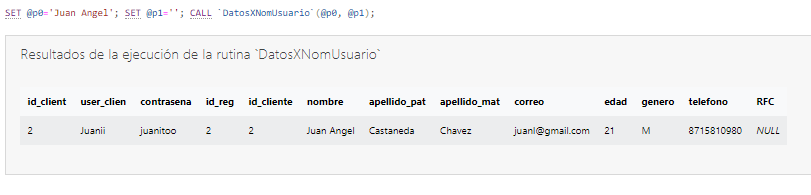


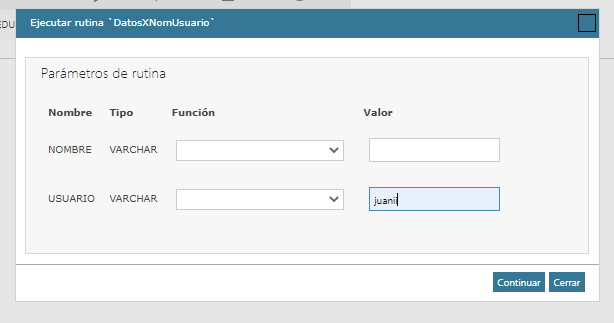
1. Mostrar información sobre los clientes por determinado nombre o usuario. (Procedimiento almacenado).

create procedure DatosXNomUsuario (IN NOMBRE varchar (30), IN USUARIO varchar(20))

select \* from clientes inner join clientes\_datos\_personales on clientes\_datos\_personales.id\_cliente=clientes.id\_client where clientes.user\_clien=USUARIO or clientes\_datos\_personales.nombre=NOMBRE







1. Reporte de ventas.

select RV.orden as 'Orden de venta', RV.cliente as 'Cliente', RV.compra as 'Fecha Orden',

sum(RV.productosvendidos) as 'Productos vendidos', sum(RV.productosvendidos\*RV.monto) as 'Total' from

(select orden.id\_orden as orden, clientes.user\_clien as cliente, orden.fecha as compra,

detalle\_orden.cantidad\_prod as Productosvendidos, productos.precio as monto from orden inner join detalle\_orden on detalle\_orden.id\_orden = orden.id\_orden

inner join productos on productos.id\_producto = detalle\_orden.id\_producto

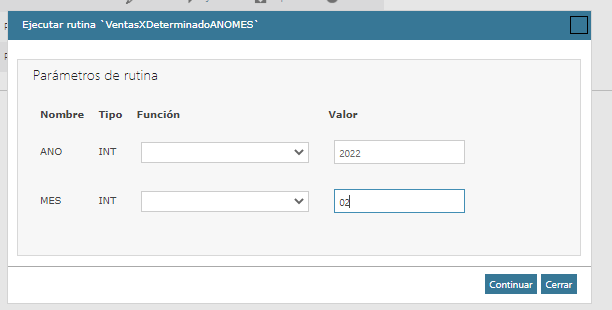
inner join clientes on clientes.id\_client = orden.id\_cliente ) as RV

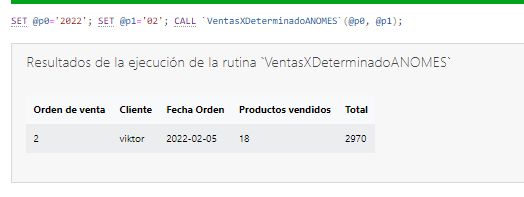
group by RV.orden;



1. Reporte de ventas por determinado mes y año (Procedimiento almacenado)

create procedure VentasXDeterminadoANOMES (IN ANO int(7), IN MES int(5)) select RV.orden as 'Orden de venta', RV.cliente as 'Cliente', RV.compra as 'Fecha Orden', sum(RV.productosvendidos) as 'Productos vendidos', sum(RV.productosvendidos\*RV.monto) as 'Total' from (select orden.id\_orden as orden, clientes.user\_clien as cliente, orden.fecha as compra, detalle\_orden.cantidad\_prod as Productosvendidos,productos.precio as monto from orden inner join detalle\_orden on detalle\_orden.id\_orden = orden.id\_orden inner join productos on productos.id\_producto = detalle\_orden.id\_producto inner join clientes on clientes.id\_client = orden.id\_cliente ) as RV WHERE month(RV.compra)=MES OR year(RV.compra)=ANO;





### *Doctor*

1. Mostrar las citas del doctor logueado



### *Cliente*

1. Mostrar los productos, el total por producto de cada uno de ellos, de un determinado cliente.

SELECT productos.nom\_producto, productos.precio, detalle\_orden.cantidad\_prod,

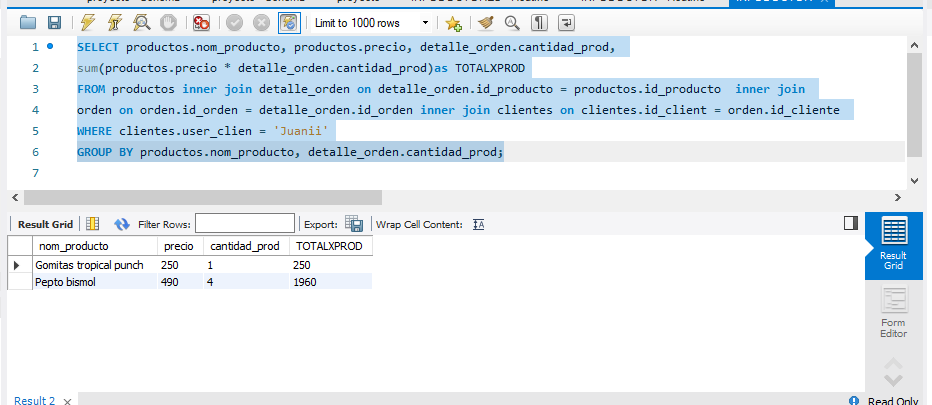
sum(productos.precio \* detalle\_orden.cantidad\_prod)as TOTALXPROD

FROM productos inner join detalle\_orden on detalle\_orden.id\_producto = productos.id\_producto inner join

orden on orden.id\_orden = detalle\_orden.id\_orden inner join clientes on clientes.id\_client = orden.id\_cliente

WHERE clientes.user\_clien = 'Juanii'

GROUP BY productos.nom\_producto, detalle\_orden.cantidad\_prod;



1. Mostrar el total de una compra de un cliente

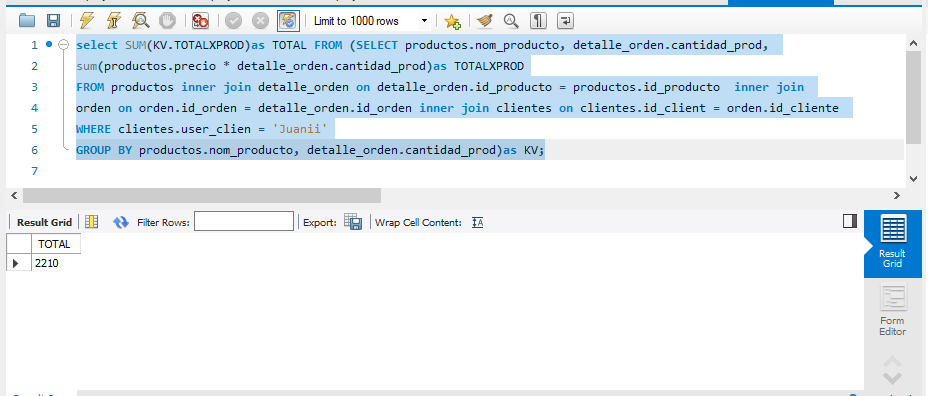
select SUM(KV.TOTALXPROD)as TOTAL FROM (SELECT productos.nom\_producto, detalle\_orden.cantidad\_prod,

sum(productos.precio \* detalle\_orden.cantidad\_prod)as TOTALXPROD

FROM productos inner join detalle\_orden on detalle\_orden.id\_producto = productos.id\_producto inner join

orden on orden.id\_orden = detalle\_orden.id\_orden inner join clientes on clientes.id\_client = orden.id\_cliente

WHERE clientes.user\_clien = 'Juanii'

GROUP BY productos.nom\_producto, detalle\_orden.cantidad\_prod)as KV;

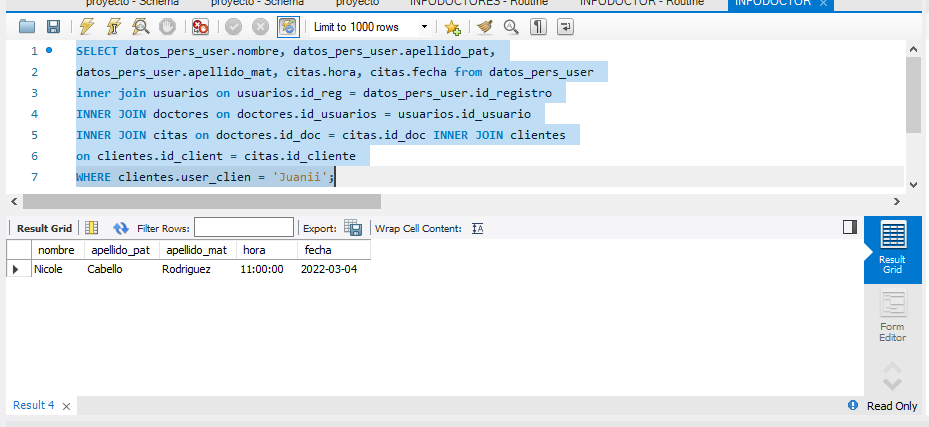
1. Mostrar las citas de un determinado cliente

SELECT datos\_pers\_user.nombre, datos\_pers\_user.apellido\_pat,

datos\_pers\_user.apellido\_mat, citas.hora, citas.fecha from datos\_pers\_user inner join usuarios on usuarios.id\_reg = datos\_pers\_user.id\_registro

INNER JOIN doctores on doctores.id\_usuarios = usuarios.id\_usuario

INNER JOIN citas on doctores.id\_doc = citas.id\_doc INNER JOIN clientes on clientes.id\_client = citas.id\_cliente WHERE clientes.user\_clien = 'Juanii';

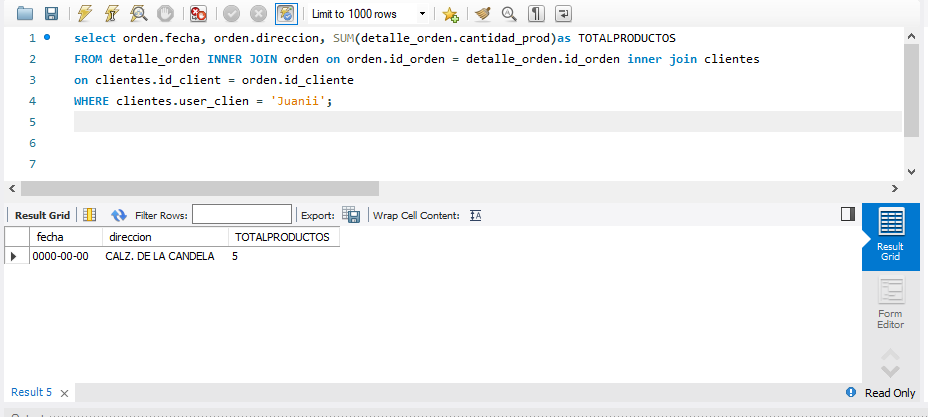


1. Mostrar los pedidos y las compras de un determinado cliente

select orden.fecha, orden.direccion, SUM(detalle\_orden.cantidad\_prod)as TOTALPRODUCTOS

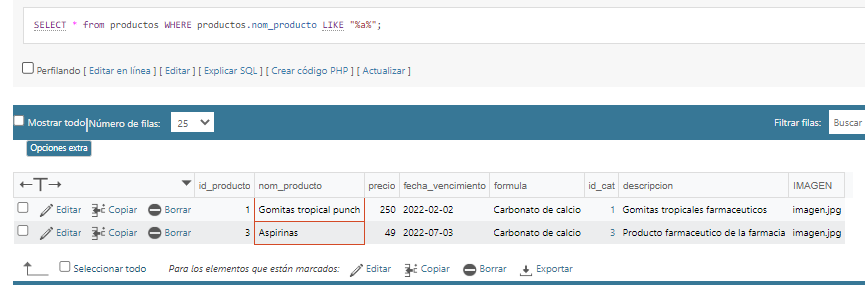
FROM detalle\_orden INNER JOIN orden on orden.id\_orden = detalle\_orden.id\_orden inner join clientes

on clientes.id\_client = orden.id\_cliente

WHERE clientes.user\_clien = 'Juanii';

1. Mostrar los productos con determinada letra o palabra

SELECT \* from productos WHERE productos.nom\_producto LIKE'%".$busqueda."%'



1. Mostrar todas las citas del cliente actual.

SELECT todo.nombre, todo.app, todo.apm, todo.especialidad,todo.cedula, todo.hora,

todo.fecha from (SELECT citas.id\_cita as id\_cita, citas.id\_doc, citas.hora as hora, citas.fecha as fecha,

datos\_pers\_user.nombre as nombre, datos\_pers\_user.apellido\_pat as app, datos\_pers\_user.apellido\_mat as apm,

doctores.especialidad as especialidad, doctores.cedula as cedula, clientes.user\_clien as cliente FROM citas

inner join clientes on clientes.id\_client=citas.id\_cliente inner join doctores on citas.id\_doc=doctores.id\_doc

inner join usuarios on usuarios.id\_usuario=doctores.id\_usuarios inner JOIN datos\_pers\_user on datos\_pers\_user.id\_registro=usuarios.id\_reg)

as todo WHERE todo.cliente='juanii';

