

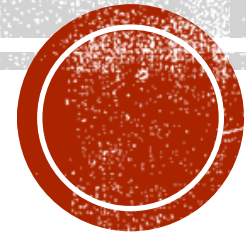
BASE DATOS DISTRIBUIDA PARA VETERINARIA

Hecho por:

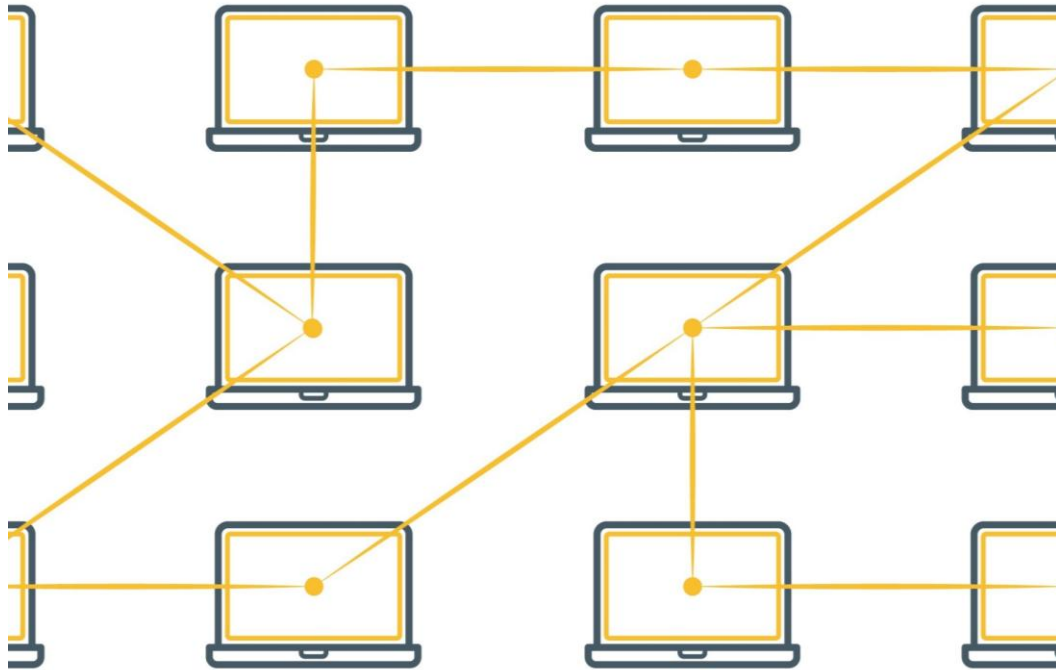
Daniel Monroy Garnica

Saul Jiménez Mercado

Alexis Ortiz Jaén



INTRODUCCIÓN



- La correcta administración de la información es esencial para ofrecer un servicio de calidad a los clientes, lo cual implica tener los registros de animales, propietarios, trabajadores, tratamientos, servicios y productos actualizados. La adopción de un sistema de bases de datos distribuidas podría ser la alternativa más adecuada para enfrentar problemas en la administración de datos.



PLANTEAMIENTO DEL PROBLEMA

- La veterinaria enfrenta desafíos en la gestión de información debido a la dispersión geográfica de sus establecimientos. Esto puede generar problemas de acceso a datos, duplicidad de información y falta de consistencia en la atención al cliente. La implementación de una base de datos distribuida resolvería estos problemas al permitir un acceso rápido y uniforme a la información en todas las sucursales



OBJETIVOS

- Objetivo general

Diseñar e implementar una base de datos distribuida que permita una gestión eficiente y homogénea de la información en todos los establecimientos de la veterinaria.

- Objetivos específicos

Identificar y definir las necesidades de la veterinaria para establecer qué tipos de datos deben ser gestionados y como deben ser accesibles.

Establecer una estructura adecuada que permita la distribución de los datos en diferentes ubicaciones físicas y garantice la integridad, disponibilidad y seguridad de la información.

Asegurar que sea capaz de sincronizar la información de manera eficiente entre todos los establecimientos de la veterinaria, garantizando que los datos se actualicen y estén disponibles de forma homogénea en todas las ubicaciones



ALCANCES Y LIMITES

■ Alcances

- Implementar en todas las sucursales de la veterinaria las bases de datos para que se pueda tener información de las mascotas sin importar a la sucursal a la que se acuda.
- Crear bases de datos homogéneas para que el uso en cualquier sucursal sea el mismo.
- Utilizar replicas para tener la información lo más eficientemente posible.

■ Limites

- No se va a poder hacer una escalabilidad vertical ya que no se cuenta con el presupuesto necesario para obtener los recursos.



ANÁLISIS DE REQUERIMIENTOS

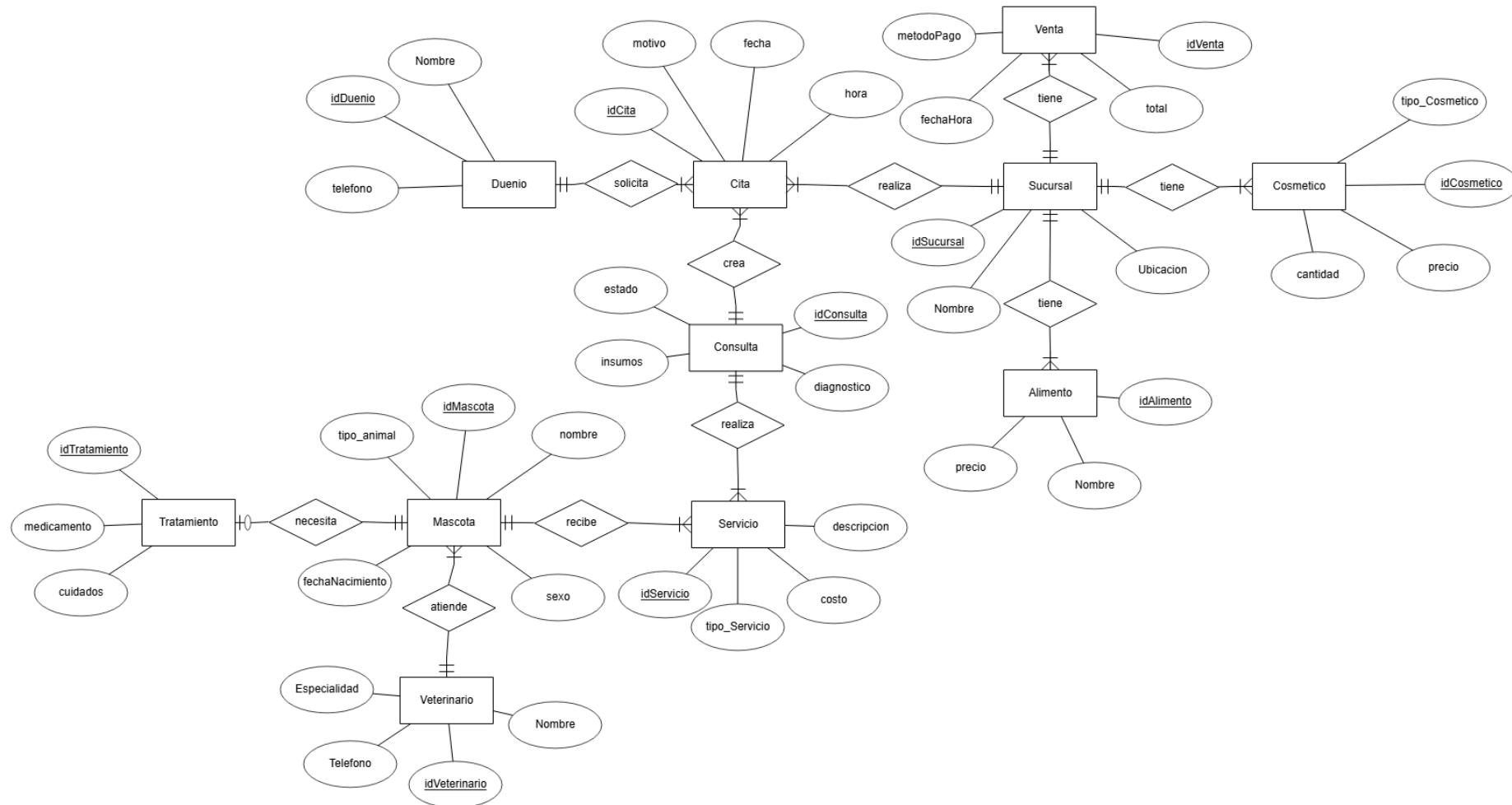
- En este sistema se busca solucionar la relación que hay entre diferentes entidades que están relacionadas con la veterinaria como lo son:

1. Mascota
2. Dueño
3. Veterinario
4. Tratamiento
5. Servicio
6. Alimento
7. Cosmético
8. Sucursal
9. Cita
10. Consulta
11. Venta

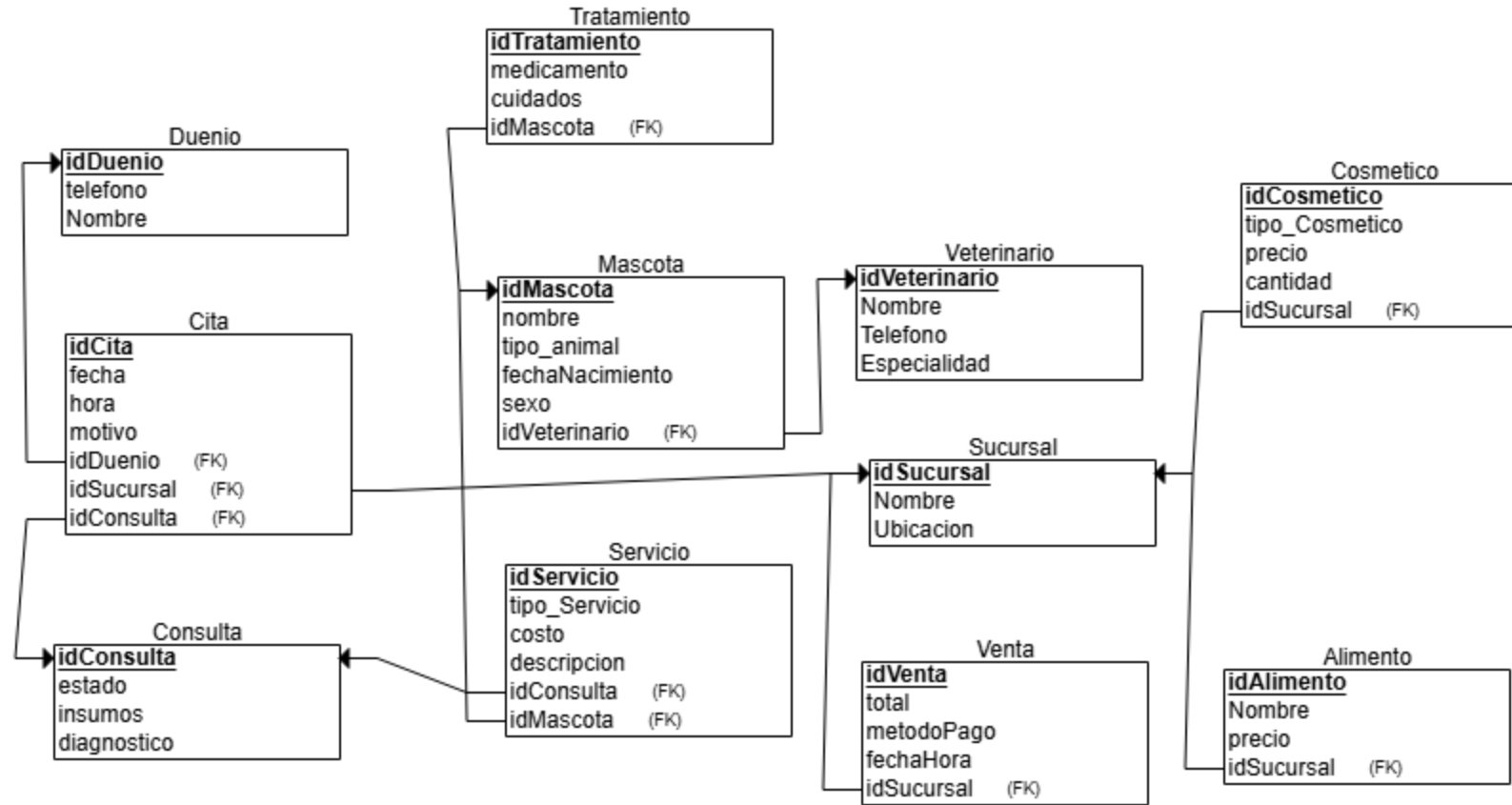


DISEÑO

■ Modelo Entidad-Relación



MODELO RELACIONAL



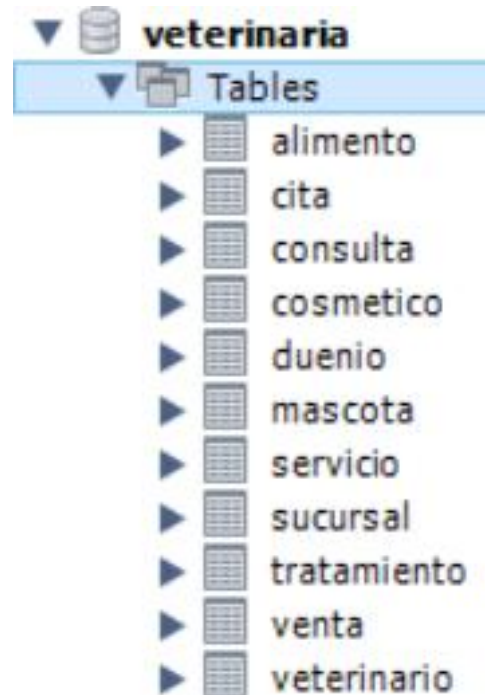
TECNOLOGÍA CLAVE

- Sistema de Base de Datos (SGBD):
 1. MySQL: Sistema de Gestión de Base de Datos (SGBD) elegido por su **fiabilidad, alto rendimiento y amplias capacidades de replicación.**
- Herramienta de Diseño/Administración:
 1. **MySQL Workbench:** Herramienta esencial para el diseño (ver Diagrama ER y Esquema Relacional en diapositivas anteriores) y la administración de las bases de datos MySQL.
- Interfaz de Acceso (API):
 1. **API (desarrollada con Visual Studio Code):** Actúa como la capa de acceso segura a la base de datos distribuida, garantizando la consistencia y el control de las operaciones realizadas desde cualquier sucursal.



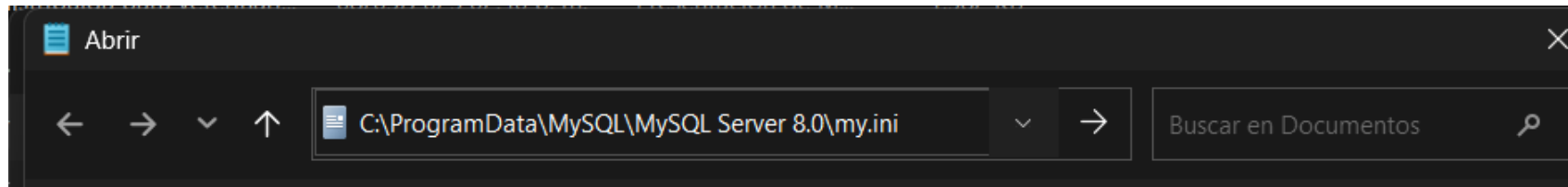
DESARROLLO

- Lo primero fue crear las tablas con las cuales se van a trabajar para la veterinaria.

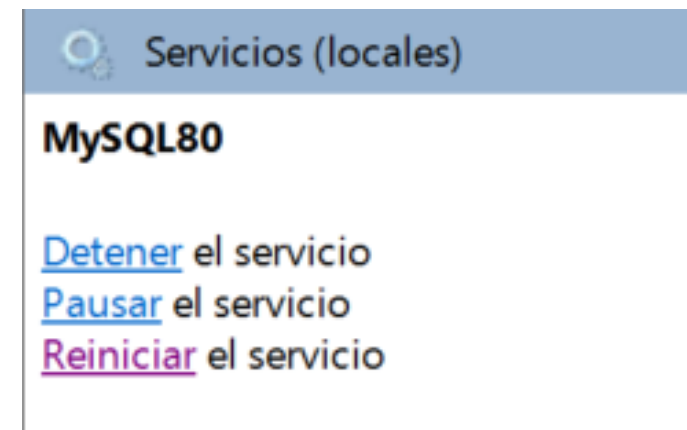


MODIFICACIÓN DE MY.INI

- Después de eso tuvimos que hacer modificaciones en un archivo de MySQL llamado my.ini para que se puedan aceptar conexiones desde cualquier IP. Se tiene que modificar desde el bloc de notas y poner bind-address =0.0.0.0 después se guardan los cambios y se reinicia el servicio de MySQL 80.

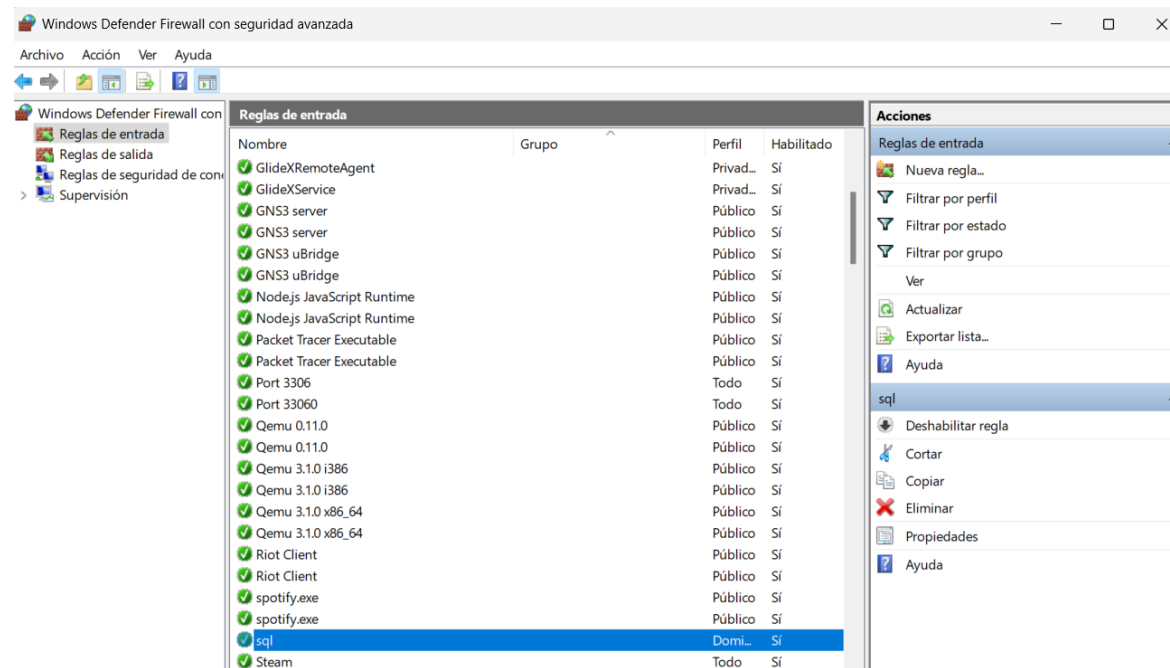


```
# SERVER SECTION
# -----
#
# The following options will be read by the MySQL Server. Make sure that
# you have installed the server correctly (see above) so it reads this
# file.
#
[mysqld]
bind-address = 0.0.0.0
```



FIREWALL

- Con esta parte ya hecha se debe de crear una regla en el firewall con la que la computadora va a permitir que se pueda manejar todas las bases de datos de forma remota y así que sea de manera mas segura las conexiones.



MODEM

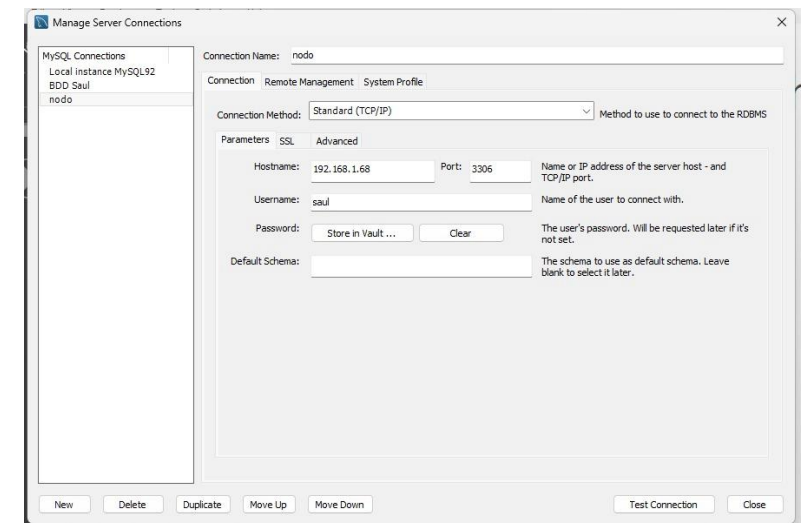
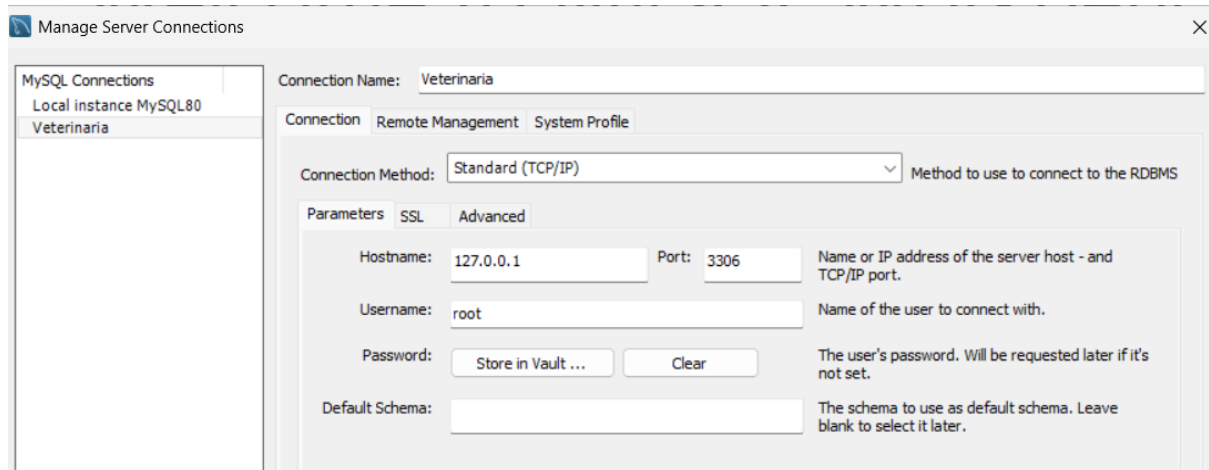
- Por medio de un modem realizamos red LAN a la que van a estar conectadas las computadoras.



VERIFICACIÓN DE IP Y CAMBIOS EN MYSQL

- Luego verificamos que ip tiene la computadora para saber como configurar el host. En la primera computadora los valores nos las da por defecto ya que va a ser el host y en el segundo le damos la ip del host, en ambos debemos de poner las credenciales correspondientes para poder acceder de manera remota.

Dirección IPv4. : 192.168.1.68



PERMISOS EN MYSQL

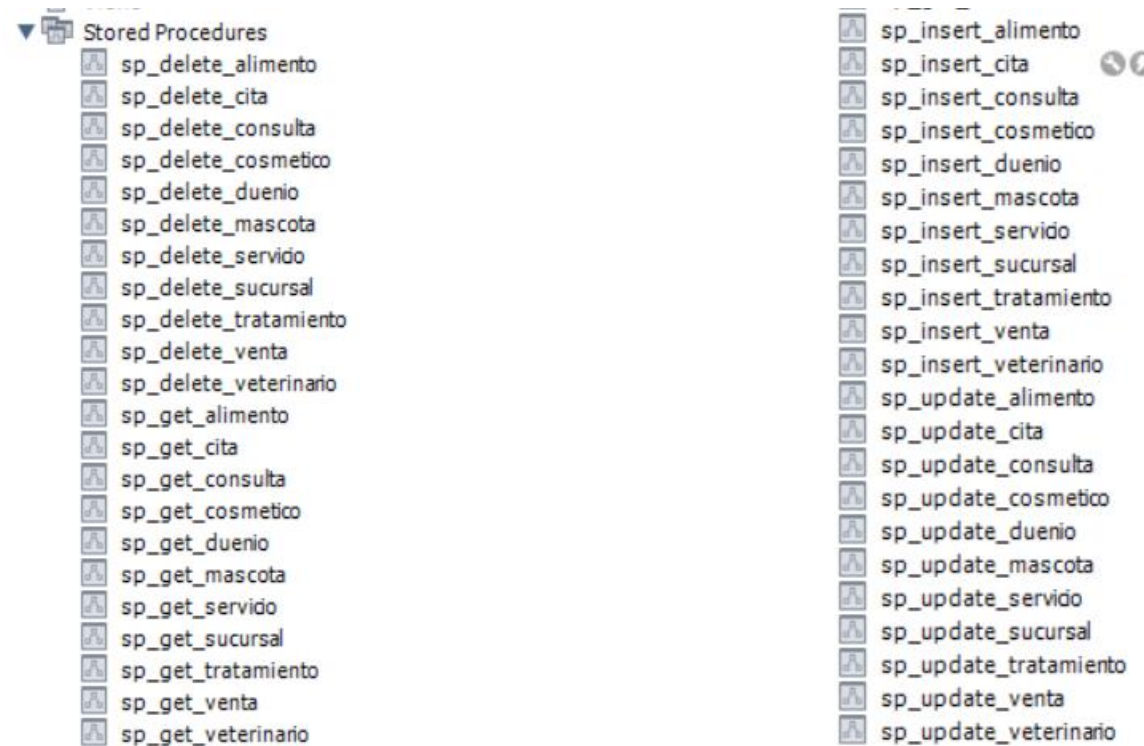
- Se deben de insertar el nombre de usuario, su dirección ip y la contraseña que va a aceptar para que la otra computadora pueda acceder a la base de datos y que si se realizan cambios se pueda ver en ambas computadoras.
- Se crea un nuevo usuario llamado saul que solo puede conectarse desde la dirección IP 192.168.1.64.
- Se conceden todos los privilegios sobre todas las bases de datos y tablas al usuario saul que se puede conectar desde la IP 192.168.1.64.
- Y recarga los privilegios desde las tablas del sistema. Esto hace que cualquier cambio de permisos entre en efecto de inmediato.

```
1 • CREATE USER 'saul'@'192.168.1.64' IDENTIFIED BY 'admin123';  
2 • GRANT ALL PRIVILEGES ON *.* TO 'saul'@'192.168.1.64' WITH GRANT OPTION;  
3 • FLUSH PRIVILEGES;
```



PROCEDIMIENTOS Y TRIGGERS

- También creamos varios procedimientos almacenados y triggers para el funcionamiento de la base de datos y que fuera mas eficiente.



CONFIGURACIÓN EN PYTHON

- Para juntarlo con una interfaz en python la cual va a tener que hacer una conexión por medio de la librería mysql-connector-python y así poder usar todas las tablas, los procedimientos almacenados y triggers dentro de la base de datos. Una cosa importante también es que en python se debe de especificar la IP, el usuario, la contraseña y la base de datos usados y así poder modificar la base de datos.

```
DB_CONFIG = {  
    'host': '127.0.0.1',  
    'user': 'daniel',  
    'password': 'admin123',  
    'database': 'veterinaria'  
}
```



INTERFACES

- Para las interfaces creamos 2, una para los administradores los cuales van a poder hacer las funciones de CRUD para todas las tablas, mientras que la otra interfaz será para los clientes y puedan realizar citas y revisar sus citas pasadas.

Veterinaria

Selecciona tabla:

idDuenio:

telefono:

nombre:

Éxito

Registro creado en Duenio.

Aceptar

Veterinaria

Selecciona tabla:

idCita:

Éxito

Registro eliminado en Cita.

Aceptar

Veterinaria

Selecciona tabla:

idCita:

fecha:

hora:

motivo:

idDuenio:

idSucursal:

idConsulta:

Éxito

Registro actualizado en Cita.

Aceptar

Veterinaria

Selecciona tabla:

idAlimento:

nombre:

precio:

idSucursal:

idAlimento	nombre	precio	idSucursal
9	Comida para conejos	18.00	5
10	Alimento para hámsters	20.00	5



CONCLUSIÓN

- En conclusión, la creación de una base de datos distribuida no es para nada sencillo ya que se deben de tener bastantes conocimientos acerca de como crear una estructura de una base de datos y también hay una gran dificultad a la hora de poder juntar una aplicación de Python con MySQL. Algunas de las mejoras que consideramos es crear mas interfaces y también dar mejores métodos de seguridad.



REFERENCIAS

- Crear un diagrama de casos de uso UML - Soporte técnico de Microsoft.(s.f.).<https://support.microsoft.com/es-es/topic/crear-un-diagrama-de-casos-de-uso-uml-92cc948d-fc74-466c-9457-e82d62ee1298>
- Rehkopf, D. M.(s.f.). Historias de usuario — Ejemplos y plantilla — Atlassian.
Atlassian.<https://www.atlassian.com/es/agile/project-management/user-stories>
- Qué es el lenguaje unificado de modelado (UML).(s.f.).
Lucidchart.<https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>
- Cornejo Velázquez, E., & Clavel Maqueda, M. (2023, 13 octubre). Estrategia de implementación Top-Down [Diapositivas].
<https://repository.uaeh.edu.mx/bitstream/bitstream/handle/123456789/20482/top-down.pdf?sequence=1&isAllowed=y>
- Equipo editorial de IONOS. (2019, 21 marzo). El modelo en cascada: desarrollo secuencial de software. IONOS Digital Guide. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>

