



Facultad de Ciencias del Hombre y la Naturaleza

Licenciatura en Ciencias de la Computación

Asignatura

Base de Datos II

Proyecto

Sistema de Control Veterinario

Bases de Datos

Catedrático

Ing. Criseida Mabel Beltrán

Integrantes

Cárcamo Rodríguez, Carlos Ernesto	CR01133496
Hernandez Vásquez, Josué Enmanuel	HV01133587
Noyola de Cárcamo, Jacqueline Patricia	N01133495
Pérez, Alex Marvin	P01133622

San Salvador, 25 de noviembre de 2017

Tabla de contenido

INTRODUCCION	3
OBJETIVOS.	4
Objetivo General	4
Objetivos específicos	4
DESCRIPCIÓN.	5
DIAGRAMA ENTIDAD RELACION.....	6
DIAGRAMA FISICO	7
SCRIPT DE LA BASE DE DATOS	8
DICCIONARIO DE DATOS.....	16
CONSULTAS EN LA BASE DE DATOS.....	25
MANUAL DE LA APLICACIÓN	36
CONCLUSIONES	42

INTRODUCCION

El trabajo en una clinica veterinaria debe de ser llevado de una forma muy ordenada, en ella se ofrecen múltiples servicios a los pacientes que la frecuentan, es por ello que se debe llevar un buen manejo de las citas, diagnosticos, consultas, vacunas que son atendidas a cada uno de ellos.

Se pretende facilitar el proceso de ordenamiento de citas, facilitándoles un sistema para dicha veterinaria. En dicho sistema se llevará un registro de cada uno de los servicios que se prestan, desde asignar la fecha de la cita, ver los diagnosticos realizados en anteriores citas, ver vacunas aplicadas a cada paciente, asignar otros servicios entre bañarlos, realizar cortes, etc.

Esto contribuirá a mejorar la atención de los pacientes y que se puedan atender de forma ordenada y eficaz para que los pacientes queden lo mejor atendidos.

El sistema de "Control de Veterinaria", será desarrollado en framework Django utilizando la base de datos POSTGRESQL como su almacenamiento de información, para poder realizar el control efectivo de cada uno de los aspectos que se toman en cuenta para el uso del sistema.

OBJETIVOS.

Objetivo General

Automatizar los procesos de consultas y controles de las mascotas que requieren los servicios veterinarios, para brindarles un servicio mas eficiente.

Objetivos específicos

- Desarrollar un sistema en Django, que actualmente se llevan de forma manual.
- Ordenar las citas, pudiendo asignarlos por fecha y por paciente.
- Crear un catálogo de pacientes, clientes y médicos que pueda ser actualizado por los usuarios autorizados.
- Crear un catálogo de servicios, para lograr una mayor rapidez al momento de asignar las citas a los paciente.

DESCRIPCIÓN.

El sistema de control veterinario es necesario para llevar un buen manejo de la información de los registros de consultas de los pacientes, desde que un paciente visita por primera vez la veterinaria, como el que hace citas para tomar los diferentes servicios que se prestan. Contará con un historial de cada una de las consultas o servicios registrados por paciente.

La información de los pacientes es almacenada definiendo sus principales características, ya que, al ser mascotas, es la forma de identificarlos, tales como raza, tamaño, sexo, peso, etc. Además, cada una de ellas se relaciona con los clientes o propietarios. De estos almacenaremos toda la información necesaria para poder comunicarnos con ellos, tales como teléfono, correo electrónico, así como su identificación por medio del dui, y sus demás datos personales.

El sistema tendrá la información de los médicos que prestan servicio a la veterinaria, tomando en cuenta todos sus datos personales, incluyendo su especialidad médica para poder ser asignados a cada paciente, dependiendo del tipo de padecimiento que presenten, como del personal asignado para diferentes tipos de servicios.

Además, contará con un control de citas para consultas de pacientes, en el que se asignaran cualquiera de los servicios que soliciten, es de aclarar que se puede dar más de un servicio en una consulta, los cuales estarán almacenados en un catálogo de servicios.

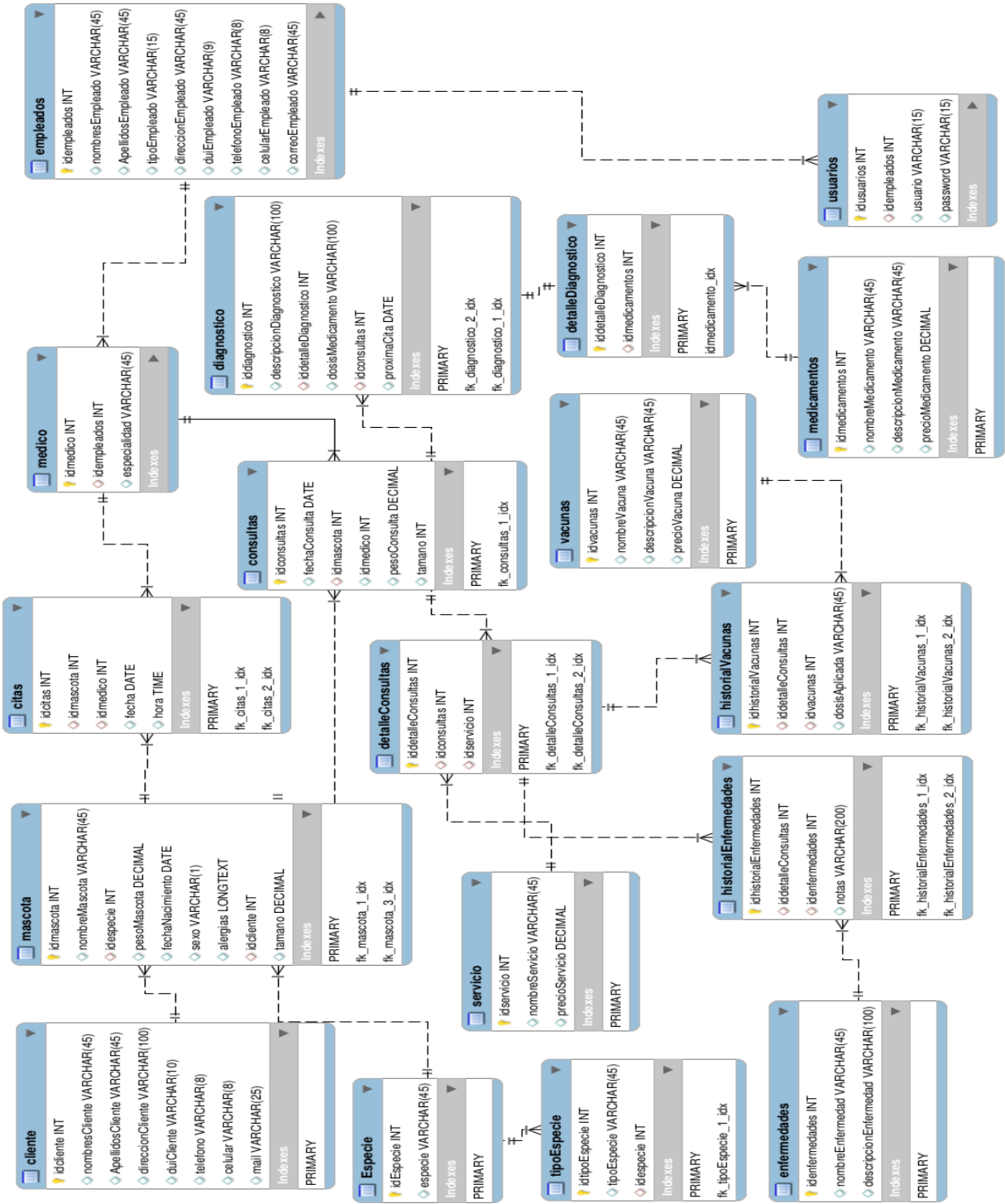
Para dar un mejor servicio, los médicos contarán con un catálogo de enfermedades para asignarlas a las consultas de los pacientes, además con un catálogo de medicamentos que serán relacionados con los diagnósticos de cada una de las consultas, toda esta información estará disponible y deberá ser actualizada periódicamente.

En el caso de servicios especiales, tales como las vacunas se llevará un registro del historial de vacunas, además de un catálogo de las mismas, con su descripción y principales características.

DIAGRAMA ENTIDAD RELACION



DIAGRAMA FISICO



SCRIPT DE LA BASE DE DATOS

-- Table Especie

```
CREATE TABLE IF NOT EXISTS Especie (  
  idEspecie INT NOT NULL,  
  especie VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (idEspecie));
```

-- Table tipoEspecie

```
CREATE TABLE IF NOT EXISTS tipoEspecie (  
  idtipoEspecie INT NOT NULL,  
  tipoEspecie VARCHAR(45) NULL DEFAULT NULL,  
  idespecie INT NULL DEFAULT NULL,  
  PRIMARY KEY (idtipoEspecie),
```

```
  CONSTRAINT fk_tipoEspecie_1  
    FOREIGN KEY (idespecie)  
    REFERENCES Especie (idEspecie)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

;

-- Table cliente

```
CREATE TABLE IF NOT EXISTS cliente (  
  idcliente INT NOT NULL,  
  nombresCliente VARCHAR(45) NULL DEFAULT NULL,  
  ApellidosCliente VARCHAR(45) NULL DEFAULT NULL,  
  direccionCliente VARCHAR(100) NULL DEFAULT NULL,  
  duiCliente VARCHAR(10) NULL DEFAULT NULL,  
  telefono VARCHAR(8) NULL DEFAULT NULL,  
  celular VARCHAR(8) NULL DEFAULT NULL,  
  mail VARCHAR(25) NULL,  
  PRIMARY KEY (idcliente))
```

;

-- Table mascota

```
CREATE TABLE IF NOT EXISTS mascota (  
  idmascota INT NOT NULL,  
  nombreMascota VARCHAR(45) NULL DEFAULT NULL,  
  idespecie INT NULL DEFAULT NULL,  
  pesoMascota DECIMAL NULL DEFAULT NULL,  
  fechaNacimiento DATE NULL DEFAULT NULL,  
  sexo VARCHAR(1) NULL DEFAULT NULL,  
  alergias varchar(300) NULL DEFAULT NULL,  
  idcliente INT NULL DEFAULT NULL,  
  tamano DECIMAL NULL,  
  PRIMARY KEY (idmascota),  
  
  CONSTRAINT fk_mascota_1  
    FOREIGN KEY (idespecie)  
    REFERENCES Especie (idEspecie)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT fk_mascota_3  
    FOREIGN KEY (idcliente)  
    REFERENCES cliente (idcliente)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
;
```

-- Table servicio

```
CREATE TABLE IF NOT EXISTS servicio (  
  idservicio INT NOT NULL,  
  nombreServicio VARCHAR(45) NULL DEFAULT NULL,  
  precioServicio DECIMAL NULL DEFAULT NULL,  
  PRIMARY KEY (idservicio))  
;
```

-- Table empleados

```
-----  
CREATE TABLE IF NOT EXISTS empleados (  
  idempleados INT NOT NULL,  
  nombresEmpleado VARCHAR(45) NULL,  
  ApellidosEmpleado VARCHAR(45) NULL,  
  tipoEmpleado VARCHAR(15) NULL,  
  direccionEmpleado VARCHAR(45) NULL,  
  duiEmpleado VARCHAR(9) NULL,  
  telefonoEmpleado VARCHAR(8) NULL,  
  celularEmpleado VARCHAR(8) NULL,  
  correoEmpleado VARCHAR(45) NULL,  
  PRIMARY KEY (idempleados))  
;
```

-- Table medico

```
-----  
CREATE TABLE IF NOT EXISTS medico (  
  idmedico INT NOT NULL,  
  idempleados INT NULL,  
  especialidad VARCHAR(45) NULL,  
  PRIMARY KEY (idmedico),  
  
  CONSTRAINT fk_medico_1  
    FOREIGN KEY (idempleados)  
    REFERENCES empleados (idempleados)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
;
```

-- Table consultas

```
-----  
CREATE TABLE IF NOT EXISTS consultas (  
  idconsultas INT NOT NULL,  
  fechaConsulta DATE NULL DEFAULT NULL,  
  idmascota INT NULL DEFAULT NULL,  
  idmedico INT NULL DEFAULT NULL,
```

```
pesoConsulta DECIMAL NULL DEFAULT NULL,  
tamano INT NULL,  
PRIMARY KEY (idconsultas),
```

```
CONSTRAINT fk_consultas_1  
  FOREIGN KEY (idmascota)  
  REFERENCES mascota (idmascota)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT fk_consultas_2  
  FOREIGN KEY (idmedico)  
  REFERENCES medico (idmedico)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```

```
;
```

```
-----  
-- Table detalleConsultas  
-----
```

```
CREATE TABLE IF NOT EXISTS detalleConsultas (  
  iddetalleConsultas INT NOT NULL,  
  idconsultas INT NULL DEFAULT NULL,  
  idservicio INT NULL DEFAULT NULL,  
  PRIMARY KEY (iddetalleConsultas),
```

```
CONSTRAINT fk_detalleConsultas_1  
  FOREIGN KEY (idconsultas)  
  REFERENCES consultas (idconsultas)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT fk_detalleConsultas_2  
  FOREIGN KEY (idservicio)  
  REFERENCES servicio (idservicio)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```

```
;
```

```
-----  
-- Table medicamentos
```

```
-----  
CREATE TABLE IF NOT EXISTS medicamentos (  
  idmedicamentos INT NOT NULL,  
  nombreMedicamento VARCHAR(45) NULL DEFAULT NULL,  
  descripcionMedicamento VARCHAR(45) NULL DEFAULT NULL,  
  precioMedicamento DECIMAL NULL DEFAULT NULL,  
  PRIMARY KEY (idmedicamentos))  
;
```

```
-----  
-- Table vacunas  
-----
```

```
CREATE TABLE IF NOT EXISTS vacunas (  
  idvacunas INT NOT NULL,  
  nombreVacuna VARCHAR(45) NULL DEFAULT NULL,  
  descripcionVacuna VARCHAR(45) NULL DEFAULT NULL,  
  precioVacuna DECIMAL NULL DEFAULT NULL,  
  PRIMARY KEY (idvacunas))  
;
```

```
-----  
-- Table historialVacunas  
-----
```

```
CREATE TABLE IF NOT EXISTS historialVacunas (  
  idhistorialVacunas INT NOT NULL,  
  iddetalleConsultas INT NULL DEFAULT NULL,  
  idvacunas INT NULL DEFAULT NULL,  
  dosisAplicada VARCHAR(45) NULL DEFAULT NULL,  
  PRIMARY KEY (idhistorialVacunas),  
  
  CONSTRAINT fk_historialVacunas_1  
    FOREIGN KEY (iddetalleConsultas)  
    REFERENCES detalleConsultas (iddetalleConsultas)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT fk_historialVacunas_2  
    FOREIGN KEY (idvacunas)  
    REFERENCES vacunas (idvacunas)  
    ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION)
```

```
;
```

```
-----  
-- Table detalleDiagnostico  
-----
```

```
CREATE TABLE IF NOT EXISTS detalleDiagnostico (  
  iddetalleDiagnostico INT NOT NULL,  
  idmedicamentos INT NULL,  
  PRIMARY KEY (iddetalleDiagnostico),
```

```
  CONSTRAINT idmedicamento  
  FOREIGN KEY (idmedicamentos)  
  REFERENCES medicamentos (idmedicamentos)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```

```
;
```

```
-----  
-- Table diagnostico  
-----
```

```
CREATE TABLE IF NOT EXISTS diagnostico (  
  iddiagnostico INT NOT NULL,  
  descripcionDiagnostico VARCHAR(100) NULL DEFAULT NULL,  
  iddetalleDiagnostico INT NULL DEFAULT NULL,  
  dosisMedicamento VARCHAR(100) NULL DEFAULT NULL,  
  idconsultas INT NULL DEFAULT NULL,  
  proximaCita DATE NULL,  
  PRIMARY KEY (iddiagnostico),
```

```
  CONSTRAINT fk_diagnostico_1  
  FOREIGN KEY (iddetalleDiagnostico)  
  REFERENCES detalleDiagnostico (iddetalleDiagnostico)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT fk_diagnostico_2  
  FOREIGN KEY (idconsultas)  
  REFERENCES consultas (idconsultas)  
  ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION)
```

```
;
```

```
-----  
-- Table enfermedades  
-----
```

```
CREATE TABLE IF NOT EXISTS enfermedades (  
  idenfermedades INT NOT NULL,  
  nombreEnfermedad VARCHAR(45) NULL DEFAULT NULL,  
  descripcionEnfermedad VARCHAR(100) NULL DEFAULT NULL,  
  PRIMARY KEY (idenfermedades))
```

```
;
```

```
-----  
-- Table historialEnfermedades  
-----
```

```
CREATE TABLE IF NOT EXISTS historialEnfermedades (  
  idhistorialEnfermedades INT NOT NULL,  
  iddetalleConsultas INT NULL DEFAULT NULL,  
  idenfermedades INT NULL DEFAULT NULL,  
  notas VARCHAR(200) NULL DEFAULT NULL,  
  PRIMARY KEY (idhistorialEnfermedades),
```

```
CONSTRAINT fk_historialEnfermedades_1  
  FOREIGN KEY (iddetalleConsultas)  
  REFERENCES detalleConsultas (iddetalleConsultas)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,
```

```
CONSTRAINT fk_historialEnfermedades_2  
  FOREIGN KEY (idenfermedades)  
  REFERENCES enfermedades (idenfermedades)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```

```
;
```

```
-----  
-- Table citas  
-----
```

```

CREATE TABLE IF NOT EXISTS citas (
  idcitas INT NOT NULL,
  idmascota INT NULL,
  idempleado INT NULL,
  fecha DATE NULL,
  hora TIME NULL,
  PRIMARY KEY (idcitas),

  CONSTRAINT fk_citas_1
    FOREIGN KEY (idmascota)
    REFERENCES mascota (idmascota)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT fk_citas_2
    FOREIGN KEY (idempleado)
    REFERENCES empleados (idempleados)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
;

```

-- Table usuarios

```

CREATE TABLE IF NOT EXISTS usuarios (
  idusuarios INT NOT NULL,
  idempleados INT NULL,
  usuario VARCHAR(15) NULL,
  password VARCHAR(15) NULL,
  PRIMARY KEY (idusuarios),

  CONSTRAINT fk_usuarios_1
    FOREIGN KEY (idempleados)
    REFERENCES empleados (idempleados)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
;

```

;

DICCIONARIO DE DATOS



public schema: Tables (18)



Table - public > citas

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idcitas	integer	-1		False	
idmascota	integer	-1		True	
idempleado	integer	-1		True	
fecha	date	-1		True	
hora	time without time zone	-1		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
citas_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idcitas)

Llave Foráneas

Nombre	Keys	Reference table	Reference keys
fk_citas_1	idmascota	mascota	idmascota
fk_citas_2	idempleado	empleados	idempleados



Table - public > cliente

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idcliente	integer	-1		False	
nombrescliente	character varying	45	NULL::character varying	True	
apellidoscliente	character varying	45	NULL::character varying	True	
direccioncliente	character varying	100	NULL::character varying	True	
ducliente	character varying	10	NULL::character varying	True	
telefono	character varying	8	NULL::character varying	True	
celular	character varying	8	NULL::character varying	True	
mail	character varying	25		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
cliente_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idcliente)



Table - public > consultas

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idconsultas	integer	-1		False	
fechaconsulta	date	-1		True	
idmascota	integer	-1		True	
idmedico	integer	-1		True	
pesoconsulta	numeric	-1		True	
tamano	integer	-1		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
consultas_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idconsultas)

Llave Foráneas

Nombre	Keys	Reference table	Reference keys
fk_consultas_1	idmascota	mascota	idmascota
fk_consultas_2	idmedico	medico	idmedico



Table - public > detalleconsultas

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
iddetalleconsultas	integer	-1		False	
idconsultas	integer	-1		True	
idservicio	integer	-1		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
detalleconsultas_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (iddetalleconsultas)

Llave Foráneas

Nombre	Keys	Reference table	Reference keys
fk_detalleconsultas_1	idconsultas	consultas	idconsultas

fk_detalleconsultas_2

idservicio

servicio

Idservicio

**Table - public > detallediagnostico****Columnas**

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
iddetallediagnostico	integer	-1		False	
idmedicamentos	integer	-1		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
detallediagnostico_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (iddetallediagnostico)

Llave Foráneas

Nombre	Keys	Reference table	Reference keys
idmedicamento	idmedicamentos	medicamentos	idmedicamentos

**Table - public > diagnostico****Columnas**

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
iddiagnostico	integer	-1		False	
descripciondiagnostico	character varying	100	NULL::character varying	True	
iddetallediagnostico	integer	-1		True	
dosismedicamento	character varying	100	NULL::character varying	True	
idconsultas	integer	-1		True	
proximacita	date	-1		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
diagnostico_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (iddiagnostico)

Llave Foráneas

Nombre	Keys	Reference table	Reference keys
--------	------	-----------------	----------------

fk_diagnostico_1
fk_diagnostico_2

iddetallediagnostico
idconsultas

[detallediagnostico](#)
[consultas](#)

iddetallediagnostico
idconsultas



Table - public > empleados

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idempleados	integer	-1		False	
nombreempleado	character varying	45		True	
apellidosempleado	character varying	45		True	
tipoempleado	character varying	15		True	
direccionempleado	character varying	45		True	
duiempleado	character varying	9		True	
telefonoempleado	character varying	8		True	
celularempleado	character varying	8		True	
correoempleado	character varying	45		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
empleados_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idempleados)



Table - public > enfermedades

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idenfermedades	integer	-1		False	
nombreenfermedad	character varying	45	NULL::character varying	True	
descripcionenfermedad	character varying	100	NULL::character varying	True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
enfermedades_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idenfermedades)



Table - public > especie

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idespecie	integer	-1		False	
especie	character varying	45	NULL::character varying	True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
especie_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idespecie)



Table - public > historialenfermedades

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idhistorialenfermedades	integer	-1		False	
iddetalleconsultas	integer	-1		True	
idenfermedades	integer	-1		True	
notas	character varying	200	NULL::character varying	True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
historialenfermedades_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idhistorialenfermedades)

Llave Foráneas

Nombre	Keys	Reference table	Reference keys
fk_historialenfermedades_1	iddetalleconsultas	detalleconsultas	iddetalleconsultas
fk_historialenfermedades_2	idenfermedades	enfermedades	idenfermedades



Table - public > historialvacunas

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idhistorialvacunas	integer	-1		False	
iddetalleconsultas	integer	-1		True	
idvacunas	integer	-1		True	
dosisaplicada	character varying	45	NULL::character varying	True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
historialvacunas_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idhistorialvacunas)

Llave Foráneas

Nombre	Keys	Reference table	Reference keys
fk_historialvacunas_1	iddetalleconsultas	detalleconsultas	iddetalleconsultas
fk_historialvacunas_2	idvacunas	vacunas	idvacunas



Table - public > mascota

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idmascota	integer	-1		False	
nombremascota	character varying	45	NULL::character varying	True	
idespecie	integer	-1		True	
pesomascota	numeric	-1		True	
fechanacimiento	date	-1		True	
sexo	character varying	1	NULL::character varying	True	
alergias	character varying	300	NULL::character varying	True	
idcliente	integer	-1		True	
tamano	numeric	-1		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
mascota_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idmascota)

Llave Foráneas

Nombre	Keys	Reference table	Reference keys
fk_mascota_1	idespecie	especie	idespecie
fk_mascota_3	idcliente	cliente	idcliente



Table - public > medicamentos

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idmedicamentos	integer	-1		False	

nombremedicamento	character varying	45	NULL::character varying	True
descripcionmedicamento	character varying	45	NULL::character varying	True
preciomedicamento	numeric	-1		True

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
medicamentos_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idmedicamentos)



Table - public > medico

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idmedico	integer	-1		False	
idempleados	integer	-1		True	
especialidad	character varying	45		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
medico_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idmedico)

Llave Foráneas

Nombre	Keys	Reference table	Reference keys
fk_medico_1	idempleados	empleados	idempleados



Table - public > servicio

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idservicio	integer	-1		False	
nombreservicio	character varying	45	NULL::character varying	True	
precioservicio	numeric	-1		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
servicio_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idservicio)



Table - public > tipospecie

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idtipoespecie	integer	-1		False	
tipoespecie	character varying	45	NULL::character varying	True	
idespecie	integer	-1		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
tipoespecie_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idtipoespecie)

Llave Foráneas

Nombre	Keys	Reference table	Reference keys
fk_tipoespecie_1	idespecie	especie	idespecie



Table - public > usuarios

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idusuarios	integer	-1		False	
idempleados	integer	-1		True	
usuario	character varying	15		True	
password	character varying	15		True	

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
usuarios_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idusuarios)

Llave Foráneas

Nombre	Keys	Reference table	Reference keys
fk_usuarios_1	idempleados	empleados	idempleados



Table - public > vacunas

Columnas

Nombre	Tipo	Longitud	Valor Defecto	Nulo	Comentario
idvacunas	integer	-1		False	
nombrevacuna	character varying	45	NULL::character varying	True	

descripcionvac una	character varying	45	NULL::character varying	True
preciovacuna	numeric	-1		True

Indices

Nombre (scan, tup_read, tup_fetch)	Primary	Unique	Clustered	Size	Definition
vacunas_pkey (0, 0, 0)	True	True	False	8.0 KB	btree (idvacunas)

CONSULTAS EN LA BASE DE DATOS

1. SELECCIONAR MASCOTAS POR FECHA DE NACIMIENTO

Select idmascota, nombremascota from mascota

where fechanacimiento BETWEEN CAST ('2005-12-05' AS DATE) AND CAST ('2005-12-07' AS DATE)

2. VER UNION DE VACUNAS CON ENFERMEDADES

Select idvacunas, nombrevacuna, descripcionvacuna from vacunas

Union

Select idenfermedades, nombreenfermedad, descripcionenfermedad from enfermedades

3. EMPLEADOS QUE SEAN MEDICOS CUYO NOMBRE INICIE CON LA LETRA "E"

SELECT e.idempleados, nombreempleado, apellidoempleado, direccionempleado, celularepleado, especialidad

FROM empleados e

join

medico m

on e.idempleados=m.idempleados

and chr(ascii(nombreempleado))='E' ;

4. BUSQUEDA DE INFORMACION DE CLIENTES POR MEDIO DE LAS MASCOTAS RELACIONADAS.

SELECT idmascota, nombremascota , nombrescliente,apellidoscliente,direccioncliente

FROM mascota m

join

cliente c

on c.idcliente=m.idcliente

WHERE nombremascota IN ('Firulay', 'Pisinga')

5. BUSQUEDA DE SERVICIOS QUE INICIEN CON LA LETRA "b"

```
select idservicio, nombreservicio, precioservicio from servicio
where nombreservicio LIKE 'b%';
```

6. VERIFICAR CITAS POR MASCOTA

```
sselect fecha as "Fecha Cita", nombremascota,especie,nombrescliente || " " || apellidoscliente as
"Cliente",nombreempleado || " " || apellidosempleado as "Empleado que registro cita"
from citas c
join
mascota m
on c.idmascota=m.idmascota
join
empleados e
on c.idempleados=e.idempleados
join
cliente cl
on m.idcliente=cl.idcliente
join
especie es
on m.idespecie=es.idespecie
where c.idmascota=1
```

7. VERIFICAR CONSULTAS POR SERVICIO

```
select fechaconsulta,nombremascota,nombreempleado || " " || apellidoempleado as "Medico",pesoconsulta as
"peso en lb",c.tamano as "tamano en cms",nombreservicio as servicio

from

consultas c

join

medico m

on c.idmedico=m.idmedico

join

empleados e

on m.idempleados=e.idempleados

join

detalleconsultas d

on d.idconsultas=c.idconsultas

join

servicio s

on s.idservicio=d.idservicio

join

mascota me

on me.idmascota=c.idmascota

where nombreservicio='Peluqueria'
```

8. VERIFICAR MASCOTAS Y ENFERMEDADES PADECIDAS

```
select nombremascota as "Mascota", nombreenfermedad as "Nombre de la Enfermedad" from mascota m
join
consultas c
on c.idmascota=m.idmascota
join
detalleconsultas d
on c.idconsultas=d.idconsultas
join
historialenfermedades h
on d.iddetalleconsultas=h.iddetalleconsultas
join
enfermedades e
on e.idenfermedades=h.idenfermedades
```

9. BUSQUEDA DE MEDICOS POR LUGAR DE RESIDENCIA

```
select * from empleados where tipoempleado='Medico' and direccionempleado like 'San%'
```

10. BUSQUEDA DE ESPECIES POR NOMBRE DE ESPECIE

```
select especie, tipoespecie from
especie e
join
tipoespecie t
using (idespecie)
where especie like 'Perr%'
```

Previous queries

```
--> CONSULTA #1  
-->SELECCIONAR MASCOTAS POR FECHA DE NACIMIENTO  
  
Select idmascota, nombremascota from mascota  
where fechanacimiento BETWEEN CAST ('2005-12-05' AS DATE) AND CAST ('2005-12-07' AS DATE)
```

Output pane

Data Output Explain Messages History

	idmascota integer	nombremascota character varying(45)
1	1	Firulay
2	2	Pisinga
3	3	domado

Previous queries

```
--> CONSULTA #2
-->VER UNION DE VACUNAS CON ENFERMEDADES
Select idvacunas, nombrevacuna, descripcionvacuna from vacunas
Union
Select idenfermedades, nombreenfermedad, descripcionenfermedad from enfermedades
```

Output pane

	idvacunas integer	nombrevacuna character varying(45)	descripcionvacuna character varying
1	2	Polivalente	a las 8 semanas
2	5	Rabia	Total
3	3	Moquillo	Mocuidad
4	1	Gripe	Calentura
5	3	hepatitis	a las 12 semanas
6	4	parvovirus	Paracitos
7	2	Tos	Malestar
8	1	Primovacunación	al nacer
9	5	parainfluenza	Gripe
10	4	Palvuvirus	Virus

Previous queries

```
--> CONSULTA #3
-->EMPLEADOS QUE SEAN MEDICOS CUYO NOMBRE INICIE CON LA LETRA "E"
SELECT e.idempleados, nombreempleado, apellidoempleado, direccionempleado, celularepleado, especialidad
FROM empleados e
join
medico m
on e.idempleados=m.idempleados
and chr(ascii(nombreempleado))='E' ;
```

Output pane

	idempleados integer	nombreempleado character varying(45)	apellidoempleado character varying(45)	direccionempleado character varying(45)	celularepleado character varying(8)	especialidad character varying(45)
1	6	Eddie Osiel	Gil	San Salvador	77779999	cirujano
2	8	Exequiel Antonio	Irigoyen	San Salvador	77777777	ortopeda

Previous queries

```
--> CONSULTA #4
-->BUSQUEDA DE INFORMACION DE CLIENTES POR MEDIO DE LAS MASCOTAS RELACIONADAS.
SELECT idmascota, nombremascota , nombrescliente,apellidoscliente,direccioncliente
FROM mascota m
join
cliente c
on c.idcliente=m.idcliente
WHERE nombremascota IN ('Firulay', 'Pisinga')
```

Output pane

Data Output Explain Messages History

	idmascota integer	nombremascota character varying(45)	nombrescliente character varying(45)	apellidoscliente character varying(45)	direccioncliente character varying(100)
1	1	Firulay	Alex Marvin	Perez	San Antonio Masahuat
2	2	Pisinga	Carlos Ernesto	Carcamo	San Salvador

Previous queries

```
--> CONSULTA #5  
-->BUSQUEDA DE SERVICIOS QUE INICIEN CON LA LETRA "b"  
select idservicio, nombreservicio, precioservicio from servicio  
where nombreservicio LIKE 'b%';
```

Output pane

Data Output Explain Messages History

	idservicio integer	nombreservicio character varying(45)	precioservicio numeric
1	1	baño	50.50

Previous queries

```
--> CONSULTA #6
-->VERIFICAR CITAS POR MASCOTA

select fecha as "Fecha Cita", nombremascota,especie,nombrescliente || ' ' || apellidoscliente as "Cliente",nombreempleado || ' ' ||
apellidosempleado as "Empleado que registro cita"
from citas c
join
mascota m
on c.idmascota=m.idmascota
join
empleados e
on c.idempleados=e.idempleados
join
cliente cl
on m.idcliente=cl.idcliente
join
especie es
on m.idespecie=es.idespecie
where c.idmascota=1
```

Output pane

	Fecha Cita date	nombremascota character varying(45)	especie character varying(45)	Cliente text	Empleado que registro cita text
1	2017-12-31	Firulay	Perro	Alex MarvinPerez	Alfonso AntonioErazo
2	2017-12-15	Firulay	Perro	Alex MarvinPerez	Alfonso AntonioErazo

```

Previous queries
--> CONSULTA #7
-->VERIFICAR CONSULTAS POR SERVICIO
select fechaconsulta,nombremascota,nombreempleado || ' ' || apellidoempleado as "Medico",pesoconsulta as "peso en lb",c.tamano as "tamano en cms",nombreservicio as servicio
from
consultas c
join
medico m
on c.idmedico=m.idmedico
join
empleados e
on m.idempleados=e.idempleados
join
detalleconsultas d
on d.idconsultas=c.idconsultas
join
servicio s
on s.idservicio=d.idservicio
join
mascota me
on me.idmascota=c.idmascota
where nombreservicio='Peluqueria'

```

Output pane

	fechaconsulta date	nombremascota character varying(45)	Medico text	peso en lb numeric	tamano en cms integer	servicio character varying(45)
1	2017-11-17	domado	Exequiel AntonioIrigoyen	5687	50	Peluqueria

```

Previous queries
--> CONSULTA #8
-->VERIFICAR MASCOTAS Y ENFERMEDADES PADECIDAS

select nombremascota as "Mascota", nombreenfermedad as "Nombre de la Enfermedad" from mascota m
join
consultas c
on c.idmascota=m.idmascota
join
detalleconsultas d
on c.idconsultas=d.idconsultas
join
historialenfermedades h
on d.iddetalleconsultas=h.iddetalleconsultas
join
enfermedades e
on e.idenfermedades=h.idenfermedades

```

Output pane

	Mascota character varying(45)	Nombre de la Enfermedad character varying(45)
1	Firulay	Gripe
2	Pisinga	Tos
3	domado	Moquillo
4	saltarin	Palvuvirus
5	poly	Rabia

Previous queries De

```
--> CONSULTA #9  
-->BUSQUEDA DE MEDICOS POR LUGAR DE RESIDENCIA  
select * from empleados where tipoempleado='Medico' and direccionempleado like 'San%'
```

Output pane

Data Output Explain Messages History

	idempleados integer	nombreempleado character varying(45)	apellidosempleado character varying(45)	tipoempleado character varying(15)	direccionempleado character varying(45)	duiempleado character varying(9)	telefonoempleado character varying(8)	celularempleado character varying(8)	correolempleado character varying(45)
1	6	Eddie Osiel	Gil	Medico	San Salvador	234561111	23345555	77779999	eddi@gmail.com
2	7	Jose Adalberto	Ssanchez	Medico	San Salvador	987652222	23346666	77778888	jose@gmail.com
3	8	Exequiel Antonio	Irigoyen	Medico	San Salvador	223343333	23347777	77777777	exequiel@gmail.com

Previous queries

```
--> CONSULTA #10  
-->BUSQUEDA DE ESPECIES POR NOMBRE DE ESPECIE  
select especie, tipospecie from  
especie e  
join  
tipoespecie t  
using (idespecie)  
where especie like 'Perr%'
```

Output pane

Data Output Explain Messages History

	especie character varying(45)	tipoespecie character varying(45)
1	Perro	Pastor Aleman
2	Perro	Chihuahua

MANUAL DE LA APLICACIÓN

1. MANUAL

1.



Descripción: Pantalla del Login donde cada usuario tendrá que ingresar con su usuario y contraseña

Descripción: Pantalla de Inicio (Home) donde se encuentran el menú de opciones a realizar

2.



Descripción: Pantalla de registro del cliente, donde se registra toda información necesaria e importante del cliente dando clic en botón "Guardar Información"

Descripción: Pantalla de registro de la mascota, donde se registra toda información necesaria e importante de la mascota dando clic en el botón "Guardar Información"

3.



Descripción: Pantalla de registro del médico, donde se registra toda información necesaria e importante del Medico dando clic en botón “Guardar Información”

Descripción: Pantalla donde se almacena o se agrega una cita, se ingresan la información requerida y solicitada en el formulario y para agregar la cita dar clic en el botón “Agregar Cita”

4.



Descripción: Pantalla para reservar cita, cuando un paciente llama y quiere reservar una cita brinda la información requerida y para reservar la cita se da clic en el botón “reservar cita”

Descripción: Pantalla de búsqueda de los registro de las mascotas ingresa ID o Nombre de la mascota y clic en el botón “Buscar “

5.



Descripción: Pantalla de búsqueda de medico ingrese el ID y clic en el botón “Buscar”

Descripción: Pantalla de registro de consulta donde al momento de guardar debe de verificar si no es necesario usar los botones de “Agregar Enfermedad”, “Agregar Vacunación”, “Agregar Historial Médico”, “Diagnostico” y Guardar Información

6.



Descripción: Pantalla de búsqueda de Enfermedades

Descripción: Pantalla de búsqueda de Vacunas

7.

SISTEMA DE VETERINARIA

Historial Medico y Servicios

Mascota:

Nombre de la Mascota:

Tipo de Mascota:

Tipo de Especie o Raza:

id	Consulta	Fecha	Motivo	Diagnostico	
4	12	10/05/2017	Vacunacion Miquillo	Vacuna	<input type="button" value="Ver"/>

SISTEMA DE VETERINARIA

Historial Vacunas

Mascota:

Nombre de la Mascota:

Tipo de Mascota:

Tipo de Especie o Raza:

id	Vacuna	Fecha	
4	Vacunacion Miquillo	10/05/2017	<input type="button" value="Ver"/>
8	Patita	10/05/2017	<input type="button" value="Ver"/>

Descripción: Pantalla de Historia del médico y servicio | Descripción: Pantalla de Historial de Vacunación

8.

SISTEMA DE VETERINARIA

Diagnósticos

id Consulta:

Mascota:

Descripcion del Diagnostico:

id	Medicamento	Dosis
2	CANDRYL 100MG	4 mg/kg divididos en dos tomas durante o una sola

SISTEMA DE VETERINARIA

Busqueda de Medicamentos

Medicamento:

id	Medicamento	Descripcion	
2	CANDRYL 100 MG	Antiflamatorio no esteroideo para uso canino en comprimidos	<input type="button" value="Seleccionar"/>

Descripción: Pantalla de Registro de Diagnóstico | Descripción: Pantalla de búsqueda de Medicamentos



Descripción: Pantalla para registro Medicamentos para la mascota

10.



21

★



22

★



	Descripción: Pantalla de Registro de Especie
Descripción: Pantalla de Registro de Tipo de Especie	Descripción: Pantalla de Registro de Enfermedades

11.

SISTEMA DE VETERINARIA

Vacunas

ID Vacuna:

Nombre Vacuna:

Precio:

Descripcion:

[Guardar](#)

25

SISTEMA DE VETERINARIA

Medicamentos

ID Medicamento:

Nombre de Medicamento:

Precio:

Descripcion:

[Guardar](#)

26

SISTEMA DE VETERINARIA

Servicios

ID Servicio:

Nombre Servicio:

Precio:

Descripcion:

[Guardar](#)

SISTEMA DE VETERINARIA

Agregar Empleado

Nombre:

Apellidos:

Tipo Empleado:

Direccion:

DUI:

Telefono:

Celular:

Correo:

[Guardar informacion](#)

Descripción: Pantalla de Registro de Vacunas	Descripción: Pantalla de Registro de Medicamentos
Descripción: Pantalla de Registro de Servicios	Descripción: Pantalla de Registro de Empleados

12.

SISTEMA DE VETERINARIA

Nuevo Usuario

id empleado:

Nombres:

Apellidos:

Direccion:

DUI:

usuario:

contraseña:

[Guardar informacion](#)

SISTEMA DE VETERINARIA

Listado Empleados

Empleado:

id	Nombres	Apellidos	tipo empleado
Seleccionar			

Descripción: Pantalla de Registro de Usuario	Descripción: Pantalla de Lista de empleados donde van una lista detallada de cada uno de los empleados
--	--

CONCLUSIONES

- Las bases de datos constituyen el núcleo de las aplicaciones, ya sean estas de escritorio o web que involucren el manejo de información.
- El propósito final de una base de datos es responder a consultas y ejecutar transacciones para ordenar y almacenar información que es relevante para el buen funcionamiento de una empresa en general.
- Con el uso de una base de datos es posible eliminar la redundancia de información con que se trabaja y a la vez disminuir o eliminar las inconsistencias, aplicando restricciones de seguridad que constituyen una de las grandes cualidades de un sistema de este tipo.
- En cuanto al diseño y creación de base datos, existen diferentes formas de organizar la información y representar relaciones entre las tablas que la conforman para mantener la integridad de la información almacenada.
- Con la existencia de una base de datos, nos damos cuenta que existen tres actores en el uso y mantenimiento de la misma:
 - El diseñador de la base de datos
 - El desarrollador que implementa las restricciones y las transacciones.
 - El usuario final que edita y consulta los datos almacenados.