# El parque zoo Santafé "parque de la conservación"

# Ivan Dario Ruiz Bernal

# 17/02/2023

# 

# SOFKA U – QA

# JUAN PINEDA

# Modelo E-R

A picture containing map, text, indoor, decorated

Description automatically generated

# Entidades:

Diagram

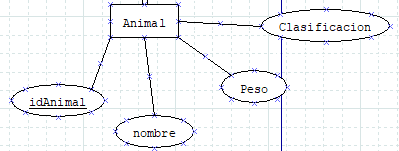
Description automatically generated

Proveedor, nos interesa guardar su información por si vemos que algún alimento llega a tener problemas podemos hacer un seguimiento preciso de dónde proviene.

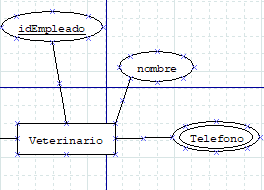
A picture containing diagram

Description automatically generated

Alimento, nos indica la dosis, el tipo y nombre de alimento, que nos sirve para saber los animales que comen, de que proveedor vienen y que veterinario los ha implementado en la dieta de los animales

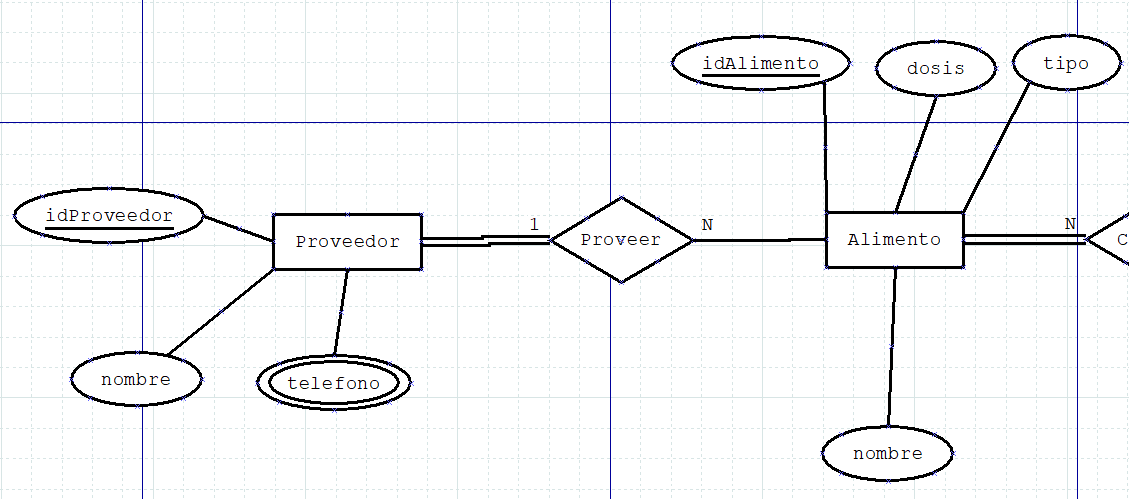


Animal, nos indica el nombre, peso y clasificación de cada animal.



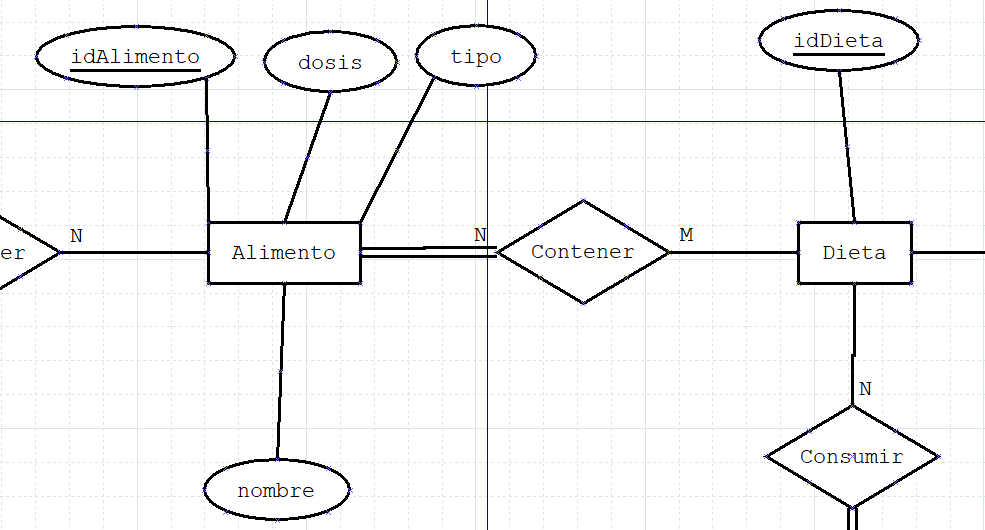
Veterinario, nos indica el nombre y los teléfonos del veterinario, además nos permite saber las dietas que ha diseñado el veterinario.

# **Relaciones:**



Primero haremos la lectura de la relación, un proveedor puede proveer uno o más alimentos, por otro lado, un alimento es proveído por un proveedor.

El alimento necesita la existencia de al menos un proveedor para existir por lo que usamos la participación total en la entidad que necesita par su existencia en la relación.



La lectura de la relación es un alimento puede estar contenido en una o más dietas, mientras una dieta puede contener uno o más alimentos.

Vemos que la dieta necesita la existencia de al menos un alimento por lo que ponemos una participación total desde la entidad necesaria hacia la relación.

Diagram

Description automatically generated

Una dieta puede ser consumida por uno o más animales, además un animal consume una o más dietas

Una dieta necesita la existencia de un animal para poder existir por lo que ponemos la participación de animal a consumir

Chart

Description automatically generated

Una dieta es diseñada por un veterinario, por otro lado, un veterinario diseña una o más dietas

Una dieta necesita la existencia de un veterinario que la diseñe para existir por lo que usamos la participación total en su entidad necesaria.

# Modelo Relacional

Diagram

Description automatically generated

Hacemos las transformaciones necesarias del modelo Entidad – Relación al modelo relacional.

Usamos el tipo de dato ROWID que nos proporciona un número de id único para cada entidad así le encargamos el identificador único a la base de datos y aseguramos la integridad de cada registro.

Lo primero que vemos es la adición de nuevas tablas, hablaremos primero de las tablas que aparecieron gracias a las relaciones.

Animal\_dieta:

Diagram

Description automatically generated

Esta tabla sale de la relación de muchos a muchos entre animal y dieta, teniendo la concatenación de las llaves primarias de las tablas mencionadas anteriormente además de ser foráneas a su vez.

Animal\_dieta:

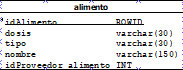


Esta tabla sale de la relación (N:M) entre alimento y dieta, teniendo como llaves principales foráneas la concatenación de las llaves principales de alimento y dieta.

Luego de tener las relaciones (N:M) transformadas a las tablas, vemos que las relaciones de uno a muchos nos generan llaves foráneas:



Vemos que tiene el id del veterinario para poder relacionarlo con las distintas dietas que pueda diseñar



Vemos que tiene el id del proveedor para poder tener una trazabilidad de la empresa que está proporcionando el alimento.

Por el momento no hemos iniciado la normalización, vemos la primera forma y vemos que cumple, ya que, tiene una clave definida y no tiene atributos multivalor ni repetidos, siendo así atributos atómicos, los atributos que veíamos multivalor se transformaron en tablas con la concatenación de su propia llave principal y la de la entidad que las contenía:





Cumple la segunda forma normal ya que, tiene una clave única ya que las columnas dependen de la llave primaria de la tabla y es única.

Cumple la tercera forma normal debido a que, cada atributo que no está incluido en la clave primaria no depende transitivamente de la clave primaria.

# Explicación de la estructura seleccionada

Elegí estas entidades y su manera de relacionarse ya que me interesa registrar el consumo de los alimentos por parte de los alimentos. Sabemos que existen personas que hacen el control del peso de los animales, por lo que los animales siempre tienen un peso, esto puede o no puede relacionarse con el alimento, por lo tanto, esta base de datos pretende ayudar a tener el control de esto.

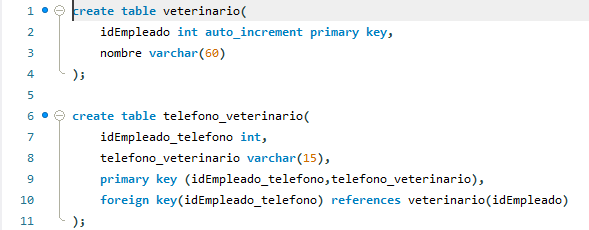
De este modo vemos a esa persona como un usuario que modifica el peso actual del animal en la base de datos, al momento de hacer esto se disparará el gatillo que guarda el peso anterior y la dieta que se le proporcionaba en un historial de pesos, así podemos tener la trazabilidad de los animales por dieta, peso, y especie. Si en dado momento el veterinario cambia la dieta también se guardará lo anterior y lo nuevo para poder tener una trazabilidad de cada especie.

Esto nos permite poder hacer un manejo de datos y un análisis exhaustivo de la mano del veterinario para ver la relación entre la comida y los pesos de los animales, por otro lado, si llega a haber una complicación con los animales y se presume que el alimento tuvo que ver con la enfermedad, se puede hacer una trazabilidad de donde viene el alimento, lo que se le daba, su dieta, su proveedor y su dosis. Esto nos permite tener un control total de la cadena de alimentación hacia los animales

No incluimos las facturas, ni las ordenes hacia el proveedor, debido a que para la funcionalidad que se le pretende dar a la base de datos no hace mucha relevancia. La factura de los alimentos proveídos no me da mucha información con respecto a que le sucede a un animal con respecto a lo que come, esta información nos serviría más si pretendiera crear una persistencia del proceso total de la gestión de los alimentos.

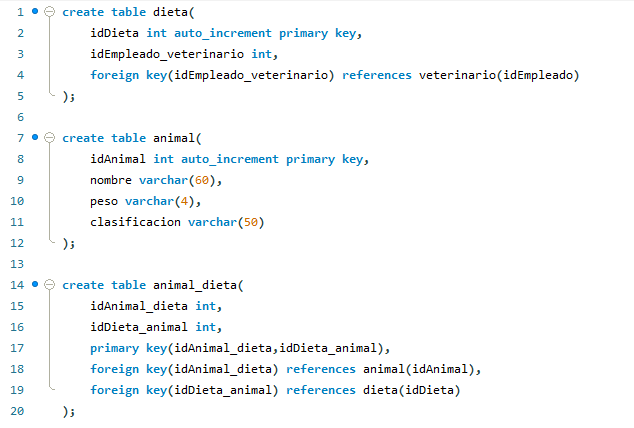
Como esta base se enfoca hacia el manejo del veterinario y sus tomas de decisiones brindamos toda la información de interés, evitando crear la persistencia de archivos que pueden ocupar espacio y no ser usados por el empleado, mientras se puede crear una base de datos de pagos que guarde las nóminas y las facturas, ya que estas se pagan a final de mes, por lo que tiene más sentido tener esta información en una base de datos de pagos que se consulte con esta regularidad.

# Creación Base de datos en SQL



Iniciamos las sentencias con las palabras reservadas “create table” las cuales nos permiten crear tablas con el nombre que indiquemos después de esta sentencia y las columnas que le pasemos dentro de los paréntesis “();”, vemos que en as columnas hay una estructura diferente, primero se indica el nombre que tendrá la columna, luego el tipo de dato que usará, en el primer caso hacemos que la base de datos asigne los valores del id por lo que ponemos la palabra reservada “auto\_increment” que hace que se cree un valor único para cada registro, y finalmente le decimos que es la llave primaria. Con el nombre solo debemos especificar su tipo.

En la segunda tabla indicamos las dos columnas solo con su tipo de dato, luego hacemos la concatenación de las columnas anteriores con la palabra reservada “primary key” y luego le indicamos que columna será una llave foránea, por lo que va a tomar datos de otra tabla, esto lo hacemos con la palabra reservada “foreign key(nombre de la tabla)”, luego le indicamos de donde se va a tomar “references” seguido del nombre de la tabla y entre paréntesis el nombre de la columna de la tabla, quedando nombre\_de\_la\_tabla(nombreColumna).



Graphical user interface, text, application, email

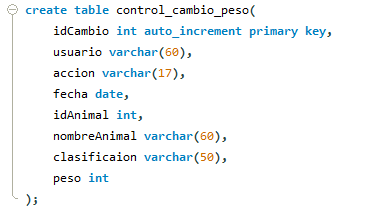
Description automatically generated

Graphical user interface, text, application

Description automatically generated

Seguimos los mismos principios explicados con la primera tabla.

# Creación de triggers y las tablas que guardan los cambios



Creamos la tabla que guardará los cambios en el peso del animal

Graphical user interface, text, application

Description automatically generated

Creamos la tabla que guardará los cambios en la dieta del animal

Graphical user interface

Description automatically generated with medium confidence

Creamos el primer trigger para guardar las actualizaciones en el peso de los animales

A picture containing graphical user interface

Description automatically generated

Creamos el segundo trigger para guardar los nuevos animales

Graphical user interface

Description automatically generated with medium confidence

Creamos el tercer trigger para guardar las actualizaciones en la tabla alimento\_dieta

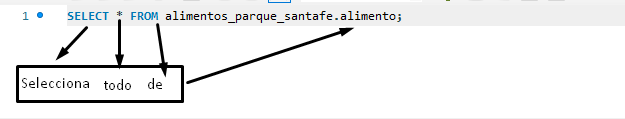
Diagram

Description automatically generated with low confidence

Creamos el cuarto trigger para guardar las dietas en el control desde que se crean

# Creación de las consultas

1. Consultar todos los alimentos:



1. Consultar todos los animales:

Diagram

Description automatically generated

1. Consultar todos los cambios e inserciones a una o más dietas:

Diagram

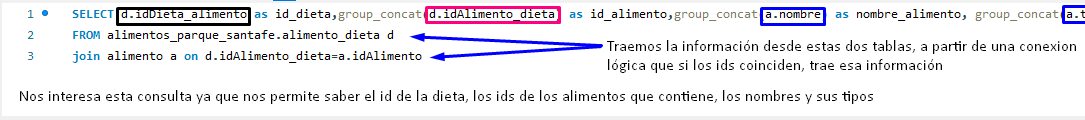
Description automatically generated

1. Consultar todos los cambios e inserciones a uno o más pesos de los animales:

Graphical user interface, text

Description automatically generated

1. Consultar la dieta de los animales:



1. Consultar proveedores:

Graphical user interface, text, application, Word

Description automatically generated

1. Consultar teléfonos de un proveedor:

Graphical user interface, text, application, email

Description automatically generated

1. Consultar Veterinario:

Graphical user interface, text, application, email

Description automatically generated

1. Consultar los teléfonos de los veterinarios:

Graphical user interface, text, application, Word

Description automatically generated

1. Consultar todos los cambios realizados por un usuario:

Graphical user interface, text, application, email

Description automatically generated

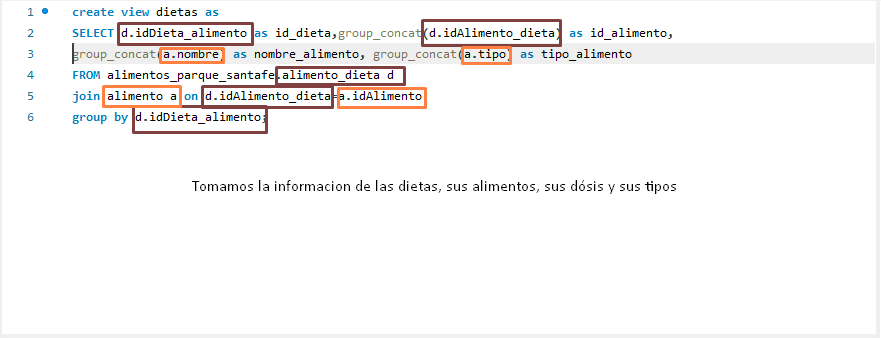
# Creación de las vistas

1.

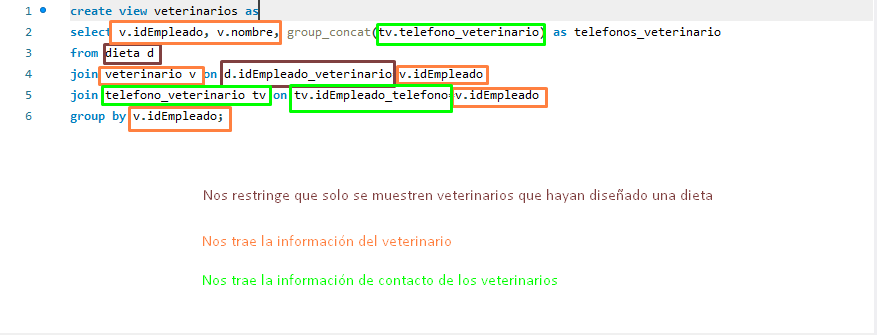
Timeline

Description automatically generated

2.



3.



4.

