

ZOOLOGICO SANTA FE

Realizado por:

Francy Julieth Ramírez Rodríguez

Presentado a:

Juan Esteban Pineda Ángel

SOFKA U

2023

ZOOLOGICO SANTA FE

Para el desarrollo del reto se me fue asignado el reto B correspondiente al Zoologico Santa fe, el cual tiene los siguientes parámetros a implementar:

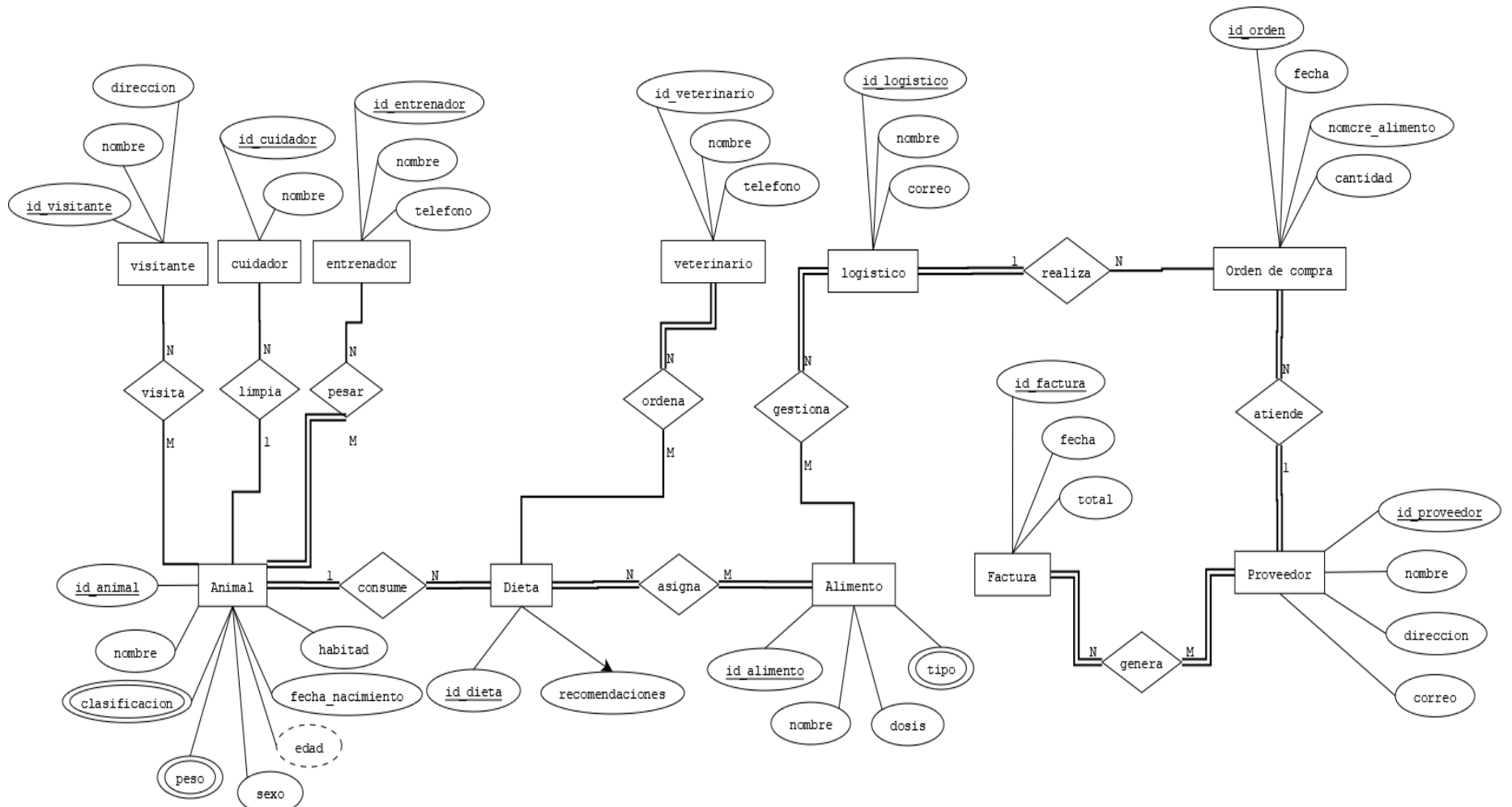
ZOO (Ejercicio B)

El parque zoo Santafé “parque de la conservación” quiere registrar en una base de datos el consumo de alimentos por los animales que tiene en su sede.

Usted acaba de hablar con el administrador y él le comenta que tienen una clasificación para los animales (mamíferos, aves, anfibios, peces y reptiles) de los cuales usted debe seleccionar 3 para el MVP.

- Tierragro empresa de alimentos para diferentes especies es uno de los 5 proveedores del parque, pero se esperan que al menos lleguen 10 nuevos proveedores.
- Dentro del parque hay varios roles para las personas, empleados cuidadores, empleados logísticos, empleados veterinarios, empleados entrenadores, visitantes.
- El veterinario esta encargada de realizar consultas a sus especies y de diseñar la dieta de cada especie.
- El alimento de cada especie es diferente y tiene una dosis y un tipo (húmeda, seca, etc).
- Uno de los roles de empleado del Zoo debe contactarse con el proveedor para solicitar alimentos y debe asear cada una de las habitas de las especies.
- El proveedor recibe una orden de compra revisa que tenga todo el alimento que le piden y con el genera una factura, a final de mes el gerente del Zoo consulta las facturas que debe a sus distintos proveedores y genera su pago correspondiente.
- Los empleados entrenadores son los encargados de llevar el peso de cada especie e informar a un veterinario en que condición están.
- El alimento es una entidad fuerte y debe contener sus características.

Teniendo en cuenta lo planteado anteriormente para el desarrollo de la actividad se procede a realizar el modelo entidad relación obteniendo:



Modelo entidad relación Zoológico Santa FE

Inicialmente se obtuvo toda la información de cada uno de los parámetros propuestos y como resultado se encontraron 11 entidades y para cada una de ellas se asignaron una serie de atributos propios de cada una, se asignaron las relaciones para cada una y su respectiva participación:

Relación de muchos a uno, este tipo de relación se encontró entre las siguientes entidades:

- Animal -Cuidador
- Animal – Dieta
- Empleado logístico – orden de compra
- Proveedor – orden de compra

Relación de uno a uno: Para este ejercicio no se encontró ninguna relación uno a uno

Relación muchos a muchos, esta relación fue asignada entre las siguientes entidades:

- Visitante – animal
- Entrenador – animal
- Veterinario – dieta
- Alimento – dieta
- Empleado logístico – alimento
- Factura – proveedor

Ya con el modelo entidad relación establecido se procede a diseñar el modelo relacional:

Para esto todas las entidades son transformadas en tablas y cada atributo en una en columnas, cada identificador único se convierte en a clave primaria de la tabla.

Para cada una de las relaciones muchos N:M se crea una nueva tabla que tendrá como clave primaria la clave primaria de las dos tablas a relacionar.

Para las relaciones 1: N la entidad del lado N añade el campo para incorporar la clave primaria de la entidad del lado 1.

Los atributos multivaluados se transformaron en nuevas tablas compuestas por la clave primaria de la entidad y el valor a agregar como atributo, ambas son asignadas como claves primarias.

visitante	cuidador	entrenador
* <u>id_visitante</u>	* <u>id_cuidador</u>	* <u>id_entrenador</u>
°nombre	°nombre	°nombre
°direccion	°id animal FK	°telefono
		°id veterinario FK

veterinario
* <u>id_veterinario</u>
◊ nombre
◊ telefono

visitante_animal
* <u>id_visitante</u>
◊ id_animal

clasificacion_animal
* <u>id_animal_clase</u>
◊ clase

entrenador_animal
* <u>id_animal</u>
* <u>id_entrenador</u>

dieta_veterinario
* <u>id_dieta</u>
* <u>id_veterinario</u>

dieta
* <u>id_dieta</u>
◊ reomendaciones
◊ id_animal FK

alimento_dieta
* <u>id_dieta</u>
* <u>id_alimento</u>

historico_peso_animal
* <u>id_animals</u>
◊ peso

logistico
* <u>id_logistico</u>
◊ nombre
◊ correo

proveedor_factura
* <u>id_proveedor</u>
* <u>id_factura</u>

alimento_logistico
* <u>id_logistico</u>
* <u>id_alimento</u>

factura
* <u>id_factura</u>
◊ fecha
◊ total

proveedor
* <u>id_proveedor</u>
◊ nombre
◊ direccion
◊ correo

alimento
* <u>id_alimento</u>
◊ nombre
◊ dosis
◊ tipo

orden_compra
* <u>id_orden</u>
◊ fecha
◊ nombre_alimento
◊ cantidad
◊ id_logistico FK
◊ id_proveedor FK

animal
* <u>id_animal</u>
◊ nombre
◊ clasificacion
◊ peso
◊ habitad
◊ edad
◊ fecha_nacimiento
◊ sexo

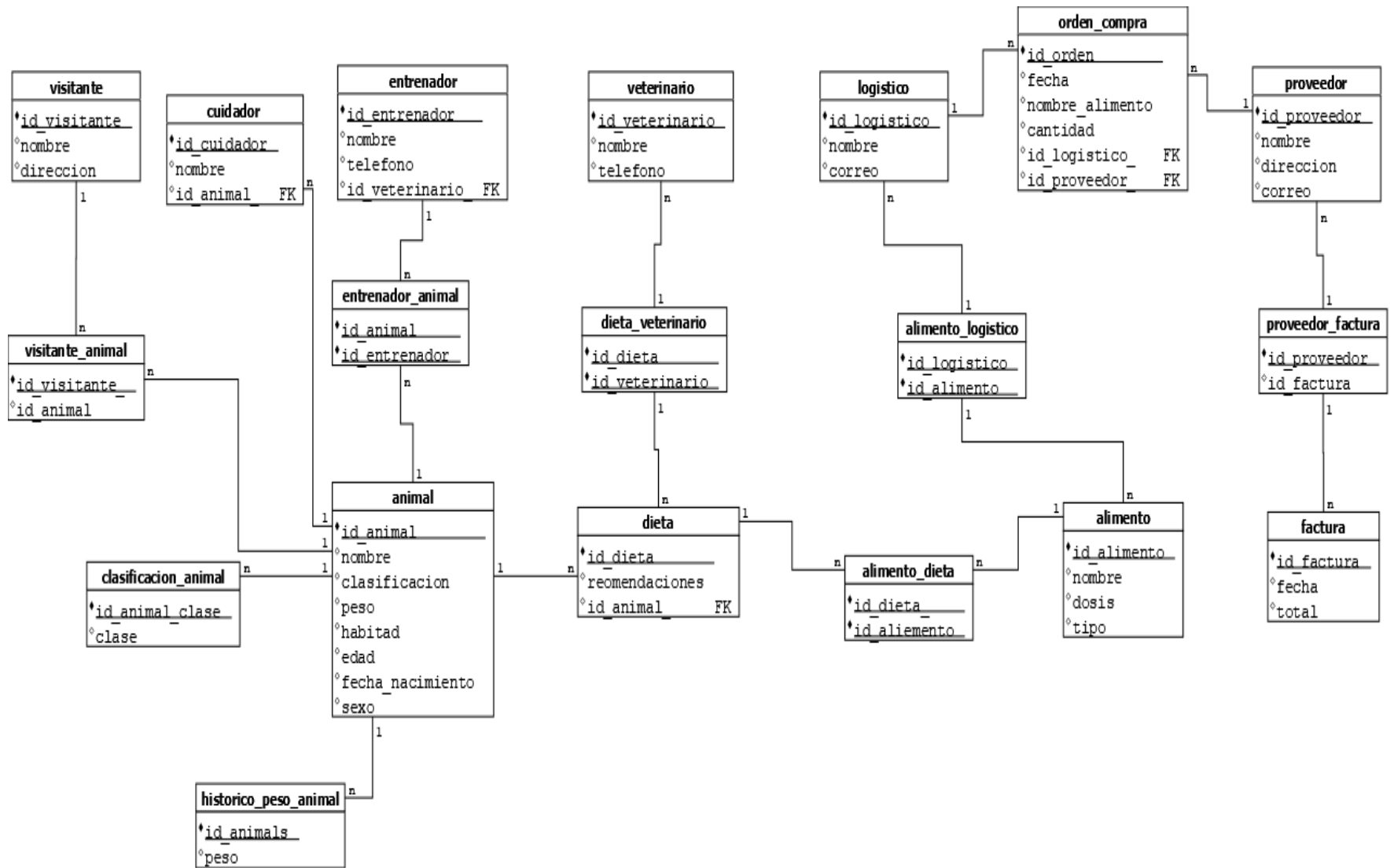
NORMALIZACION:

1NF: Con lo anterior podemos ver que todos los atributos tienen valores atómicos, no hay atributos multivaluados, no hay registros duplicados, no hay columnas duplicadas, claves principales definidas.

2FN: El esquema cumple con la primera forma normal, todos los valores de las columnas se encuentran solamente dependientes de la clave primaria de la tabla, todas las tablas tienen una clave primaria y atributos que dependen de ella.

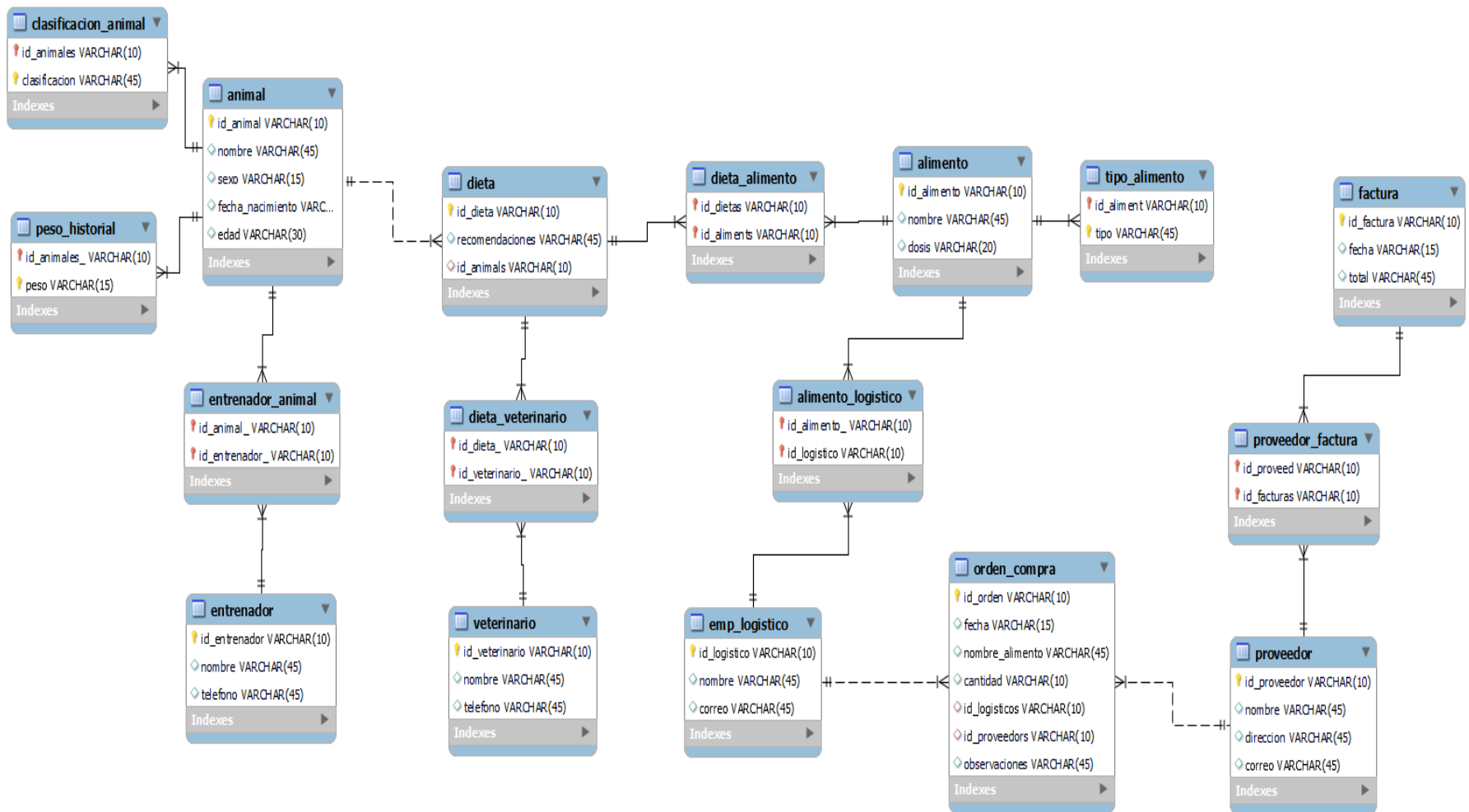
3FN: las tablas se encuentran en segunda forma normal, ningún atributo tiene dependencia transitiva, todos los atributos o campos dependen directamente de la clave principal.

Posterior a eso obtenemos:



Modelo relacional zoologico.

Teniendo en cuenta que **el objetivo principal planteado es registrar el consumo de alimentos** por animales que tiene en la sede y ya que no se especifica funcionalidad explícita a varias entidades extraídas se procede a priorizar el modelo y **se elimina la entidad visitante y cuidador**. Con esto el modelo relacional queda de la siguiente manera



Para el parámetro **uno de los roles de empleado del Zoo debe contactarse con el proveedor para solicitar alimentos** se planteo que el empleado de logística es quien gestiona los alimentos que se encuentran en el zoológico y es quien al momento de existir pocas existencias de estos procede a contactar al proveedor del alimento por medio de una orden de compra, el proveedor atiende dicha orden y realiza una factura, la cual posteriormente será visualizada a fin de mes por el gerente de la empresa. Para lograr que el gerente obtenga esta información se usara una vista.

Igualmente para el parámetro **Los empleados entrenadores son los encargados de llevar el peso de cada especie e informar a un veterinario en que condición están,** para esto el entrenador registrara periódicamente el peso del animal, este peso será guardado en una tabla que es propia del animal, la cual puede relacionar el historial de los pesos registrados por el entrenador, estos registros podrán ser visualizados por el veterinario por medio de una consulta SQL que brindara todos los datos necesarios para el veterinario. Esta información será mostrada por medio de una vista.

Ya con el modelo relacional definido se crea la base de datos por medio de sentencias SQL:

- Se crea base de datos

```
CREATE DATABASE zoologico;
```

```
USE zoologico;
```

- Se crea la tabla animal

```
CREATE TABLE animal (  
  id_animal VARCHAR(10) NOT NULL,  
  nombre VARCHAR(45) NULL,  
  sexo VARCHAR(15) NULL,  
  fecha_nacimiento VARCHAR(15) NULL,  
  edad VARCHAR(30) NULL,  
  PRIMARY KEY (id_animal));
```

- Se crea la tabla clasificación del animal

```
CREATE TABLE clasificacion_animal (  
  id_animales VARCHAR(10) NOT NULL,  
  clasificacion VARCHAR(45) NOT NULL,  
  PRIMARY KEY (id_animales, clasificacion),  
  CONSTRAINT id_animales  
  FOREIGN KEY (id_animales)
```

REFERENCES animal (id_animal));

- Se crea la tabla veterinario

```
CREATE TABLE veterinario (  
id_veterinario VARCHAR(10) NOT NULL,  
nombre VARCHAR(45) NULL,  
telefono VARCHAR(45) NULL,  
PRIMARY KEY (id_veterinario));
```

- Se crea la tabla entrenador

```
CREATE TABLE entrenador (  
id_entrenador VARCHAR(10) NOT NULL,  
nombre VARCHAR(45) NULL,  
telefono VARCHAR(45) NULL,  
id_veterinarios VARCHAR(10) NULL,  
PRIMARY KEY (id_entrenador));
```

- Se crea tabla entrenador_animal

```
CREATE TABLE entrenador_animal (  
id_animal_ VARCHAR(10) NOT NULL,  
id_entrenador_ VARCHAR(10) NOT NULL,  
PRIMARY KEY (id_animal_, id_entrenador_),  
CONSTRAINT id_animal_  
FOREIGN KEY (id_animal_)  
REFERENCES animal (id_animal),  
CONSTRAINT id_entrenador_  
FOREIGN KEY (id_entrenador_)  
REFERENCES entrenador (id_entrenador));
```

- Se crea la tabla dieta

```
CREATE TABLE dieta (  
id_dieta VARCHAR(10) NOT NULL,  
recomendaciones VARCHAR(45) NULL,  
id_animals VARCHAR(10) NULL,  
PRIMARY KEY (id_dieta),  
CONSTRAINT id_animals  
FOREIGN KEY (id_animals)  
REFERENCES animal (id_animal));
```

- Se crea la tabla dieta veterinario

```
CREATE TABLE dieta_veterinario (  
  id_dieta_ VARCHAR(10) NOT NULL,  
  id_veterinario_ VARCHAR(10) NOT NULL,  
  PRIMARY KEY (id_dieta_, id_veterinario_),  
  CONSTRAINT id_dieta_  
    FOREIGN KEY (id_dieta_)  
      REFERENCES dieta (id_dieta),  
  CONSTRAINT id_veterinario_  
    FOREIGN KEY (id_veterinario_)  
      REFERENCES veterinario (id_veterinario));
```

- Se crea la tabla alimento

```
CREATE TABLE alimento (  
  id_alimento VARCHAR(10) NOT NULL,  
  nombre VARCHAR(45) NULL,  
  dosis VARCHAR(20) NULL,  
  PRIMARY KEY (id_alimento));
```

- Se crea la tabla tipo_alimento

```
CREATE TABLE tipo_alimento (  
  id_aliment VARCHAR(10) NOT NULL,  
  tipo VARCHAR(45) NOT NULL,  
  PRIMARY KEY (id_aliment, tipo),  
  CONSTRAINT id_aliment  
    FOREIGN KEY (id_aliment)  
      REFERENCES alimento (id_alimento));
```

- Se crea la tabla dieta_alimento

```
CREATE TABLE dieta_alimento (  
  id_dietas VARCHAR(10) NOT NULL,  
  id_aliments VARCHAR(10) NOT NULL,  
  PRIMARY KEY (id_dietas, id_aliments),  
  CONSTRAINT id_dietas  
    FOREIGN KEY (id_dietas)  
      REFERENCES dieta (id_dieta),  
  CONSTRAINT id_aliments  
    FOREIGN KEY (id_aliments)  
      REFERENCES alimento (id_alimento));
```

- Se crea tabla emp_logistico

```
CREATE TABLE emp_logistic (  
  id_logistico VARCHAR(10) NOT NULL,  
  nombre VARCHAR(45) NULL,  
  correo VARCHAR(45) NULL,  
  PRIMARY KEY (id_logistico));
```

- Se crea tabla alimento_logistico

```
CREATE TABLE alimento_logistico (  
  id_alimento_ VARCHAR(10) NOT NULL,  
  id_logistico VARCHAR(10) NOT NULL,  
  PRIMARY KEY (id_alimento_, id_logistico),  
  CONSTRAINT id_alimento_  
    FOREIGN KEY (id_alimento_)  
    REFERENCES alimento (id_alimento),  
  CONSTRAINT id_logistico  
    FOREIGN KEY (id_logistico)  
    REFERENCES emp_logistic (id_logistico));
```

- Se crea tabla proveedor

```
CREATE TABLE proveedor (  
  id_proveedor VARCHAR(10) NOT NULL,  
  nombre VARCHAR(45) NULL,  
  direccion VARCHAR(45) NULL,  
  correo VARCHAR(45) NULL,  
  PRIMARY KEY (id_proveedor));
```

- Se crea la tabla orden_compra

```
CREATE TABLE orden_compra (  
  id_orden VARCHAR(10) NOT NULL,  
  fecha VARCHAR(15) NULL,  
  nombre_alimento VARCHAR(45) NULL,  
  cantidad VARCHAR(10) NULL,  
  id_logisticos VARCHAR(10) NULL,  
  id_proveedores VARCHAR(10) NULL,  
  observaciones VARCHAR(45) NULL,  
  PRIMARY KEY (id_orden),  
  CONSTRAINT id_logisticos  
    FOREIGN KEY (id_logisticos)  
    REFERENCES emp_logistic (id_logistico),  
  CONSTRAINT id_proveedores  
    FOREIGN KEY (id_proveedores)
```

REFERENCES proveedor (id_proveedor));

- Se crea la tabla factura

```
CREATE TABLE factura (  
  id_factura VARCHAR(10) NOT NULL,  
  fecha VARCHAR(15) NULL,  
  total VARCHAR(45) NULL,  
  PRIMARY KEY (id_factura));
```

- Se crea la tabla proveedor_factura

```
CREATE TABLE proveedor_factura (  
  id_proveed VARCHAR(10) NOT NULL,  
  id_facturas VARCHAR(10) NOT NULL,  
  PRIMARY KEY (id_proveed, id_facturas),  
  CONSTRAINT id_proveed  
    FOREIGN KEY (id_proveed)  
      REFERENCES proveedor (id_proveedor),  
  CONSTRAINT id_facturas  
    FOREIGN KEY (id_facturas)  
      REFERENCES factura (id_factura));
```

- Se crea la tabla peso_historial

```
CREATE TABLE peso_historial (  
  id_animales_ VARCHAR(10) NOT NULL,  
  peso VARCHAR(15) NOT NULL,  
  PRIMARY KEY (id_animales_, peso),  
  CONSTRAINT peso_animales_  
    FOREIGN KEY (id_animales_)  
      REFERENCES animal (id_animal));
```

Consultas realizadas en la base de datos:

1. MOSTRAR LISTA DE ALIMENTOS

```
select alimento.nombre, alimento.dosis, tipo_alimento.tipo  
from alimento  
inner join tipo_alimento on alimento.id_alimento = tipo_alimento.id_aliment;
```

Por medio de esta consulta vemos el nombre del alimento, la dosis y el tipo.

2. MOSTRAR LISTA DE PROVEEDORES

```
select * from proveedor;
```

Esta consulta nos permite acceder a todos los datos de cada uno de los proveedores registrados en la base de datos.

3. MOSTRAR DIETA

```
select veterinario.nombre, dieta.recomendaciones, alimento.nombre  
from veterinario  
inner join dieta_veterinario on dieta_veterinario.id_veterinario_ = veterinario.id_veterinario  
inner join dieta on dieta.id_dieta = dieta_veterinario.id_dieta_  
inner join dieta_alimento on dieta_alimento.id_dietas = dieta.id_dieta  
inner join alimento on alimento.id_alimento = dieta_alimento.id_alimentos;
```

En esta consulta podemos ver el nombre del veterinario que atendió el animal, las recomendaciones realizadas para esa dieta en específico y el nombre del alimento ordenado para la dieta.

4. Mostrar historial de peso = el veterinario podrá consultar el historial de peso por animal y notar los cambios existidos por cada registro

```
select animal.nombre, clasificacion_animal.clasificacion, peso_historial.peso  
from animal  
inner join clasificacion_animal on animal.id_animal = clasificacion_animal.id_animales  
inner join peso_historial on animal.id_animal = peso_historial.id_animales_;
```

Esta consulta se obtiene el historial del peso de los animales, cada animal tendrá una tabla para actualizar el peso cada vez que sea necesario por medio de la tabla peso_historico donde se podrá acceder. Como dentro de unos de los parámetros del ejercicio es que el entrenador registra el peso animal e informa al veterinario de su situación actual se podrá cumplir con este parámetro gracias a esta consulta, el veterinario podrá evaluar cada animal y definir si la dieta ordenada está cumpliendo con su objetivo y si es o no necesario modificar la dieta.

5. MOSTRAR LOS EMPLEADOS

```
select veterinario.nombre, entrenador.nombre, emp_logistic.nombre  
from veterinario, entrenador, emp_logistic;
```

Esta consulta nos permite acceder al listado de todos los empleados por su nombre.

6. MOSTRAR FACTURAS

```
select * from facturas;
```

Esta consulta nos permite ver el listado de las facturas existentes.

7. MOSTRAR LISTA DE ANIMALES Y SU CLASIFICACIÓN

```
select animal.nombre, clasificacion_animal.clasificacion
from animal
inner join clasificacion_animal on animal.id_animal = clasificacion_animal.id_animales;
```

Por medio de esta consulta se obtiene la lista de los animales por su nombre y a que clasificación pertenece.

8. Mostrar orden de compra

```
select * from orden_compra;
```

Se generan la lista de todas las ordenes de compra

9. Entrenador que pesa al animal

```
select entrenador.nombre, animal.nombre
from entrenador
inner join entrenador_animal on entrenador.id_entrenador =
entrenador_animal.id_entrenador_
inner join animal on animal.id_animal = entrenador_animal.id_animal_;
```

En esta consulta se permite conocer cual es el entrenador que registra el nuevo peso del animal, este registro se carga en la tabla peso_historial.

10. Alimentos gestionados por el empleado logistico

```
select alimento.nombre, alimento.dosis, emp_logistic.nombre
from alimento
inner join alimento_logistico on alimento.id_alimento = alimento_logistico.id_alimento_
inner join emp_logistic on emp_logistic.id_logistico = alimento_logistico.id_logistico;
```

Para esta consulta se traen los datos del empleado que se encarga de gestionar la compra de mas alimentos para el zoológico, se muestra cual fue el empleado que gestionó que alimento.

VISTAS

Como el planteamiento inicial especifica que el objetivo es la gestión del consumo de alimentos de los animales del ZOO la conservación se escogen las siguientes vistas en búsqueda de seguir el objetivo planteado

- **Vista 1:**

```
# VISTA 1: Permite ver la lista de los alimentos que existen en el zoologico
CREATE VIEW ListaAlimentos AS
select alimento.nombre, alimento.dosis, tipo_alimento.tipo
from alimento
inner join tipo_alimento on alimento.id_alimento = tipo_alimento.id_alimento;
SELECT * FROM ListaAlimentos;
```

Por medio de esta vista se puede obtener información relevante como es los alimentos existentes dentro del zoológico, esto permite conocer que alimentos pueden ser asignados para la dieta de cada animal, según su dosis y el tipo de alimento que sea.

- **Vista 2:**

```
#VISTA 2: Se muestra registros que corresponden a la dieta ordenada por el veterinario
CREATE VIEW DietaAnimal AS
select veterinario.nombre as Vetertinario, dieta.recomendaciones AS Recomendaciones, alimento.nombre AS Alimento_ordenado
from veterinario
inner join dieta_veterinario on dieta_veterinario.id_veterinario_ = veterinario.id_veterinario
inner join dieta on dieta.id_dieta = dieta_veterinario.id_dieta_
inner join dieta_alimento on dieta_alimento.id_dietas = dieta.id_dieta
inner join alimento on alimento.id_alimento = dieta_alimento.id_aliments;
select * from DietaAnimal;
```

En esta vista se destacan datos importantes para la gestión de consumo de alimentos dentro del zoológico, podemos ver los datos correspondientes a la dieta como el veterinario que la ordeno, que recomendaciones da para la administración del alimento, el nombre del alimento ordenado. Se considera relevante esta vista ya que se mantiene ligada al objetivo principal del ejercicio.

- **Vista 3:**

```
# VISTA 3: Mostrar historial del peso por medio de la vista
CREATE VIEW HistorialPeso AS
select animal.nombre AS NombreAnimal, clasificacion_animal.clasificacion AS ClaseAnimal,
group_concat(peso_historial.peso separator ' ,') AS PesosRegistrados
from animal
inner join clasificacion_animal on animal.id_animal = clasificacion_animal.id_animales
inner join peso_historial on animal.id_animal = peso_historial.id_animales_;
select * from HistorialPeso;
```

Esta información le permitiría al veterinario consultar el registro de los pesos de cada animal con esto puede saber si es necesario modificar la dieta y por consiguiente se gestiona los alimentos por cada animal.

- **Vista 4:**

```
# VISTA 4: Mostrar historial de empleados que han gestionado alimentos
CREATE VIEW GestionAlimentos AS
select alimento.nombre AS Alimento, alimento.dosis, emp_logistic.nombre AS EmpleadoGestor
from alimento
inner join alimento_logistico on alimento.id_alimento = alimento_logistico.id_alimento_
inner join emp_logistic on emp_logistic.id_logistico = alimento_logistico.id_logistico;
select * from GestionAlimentos;
```

Podemos llevar un registro de la gestión realizada por cada empleado encargado de mantener la disponibilidad de los alimentos para el zoológico, de esta gestión depende que los animales siempre tengan un alimento disponible para su consumo, el empleado garantiza que haya alimentos para cada dieta.

TRIIGGER

Se crea la tabla que permite registrar cada una de las acciones que se activaran al realizar alguna de las acciones establecidas para cada trigger, donde se carga el nombre del usuario que realiza la acción, acción realizada y la fecha y hora de la novedad.

```

# CREACION DE LA TABLA control_de_cambios_zoologico
CREATE TABLE control_de_cambios_zoologico (
  usuario VARCHAR(45),
  accion VARCHAR (45),
  fecha DATETIME DEFAULT current_timestamp
);

```

1. TRIGGER registro_agregar_peso

```

# 1. CREACION DE TRIGGER PARA REGISTRO DE ACCION AL AGREGAR PESO
DELIMITER //
CREATE TRIGGER registro_agregar_peso
AFTER INSERT
ON peso_historial FOR EACH ROW
BEGIN
  INSERT INTO control_de_cambios_zoologico (usuario,accion,fecha)
  VALUES ('Sergio Paez, entrenador','Agregar peso',now());
END
//
#Se usa el procedimiento agregar peso para cargar el registro
CALL agregarPeso('1','1600 gtramos');

```

Por medio de este trigger se puede llevar un control sobre el veterinario que realiza el registro de un nuevo peso del animal, conociendo así la fecha y hora del registro y de igual manera quien lo realizo.

2. TRIGGER registro_borrar_proveedor

```

# 2. CREACION DE TRIGGER PARA REGISTRO DE ACCION AL ELIMINAR UN PROVEEDOR
DELIMITER //
CREATE TRIGGER registro_borrar_proveedor
AFTER DELETE
ON proveedor FOR EACH ROW
BEGIN
  INSERT INTO control_de_cambios_zoologico (usuario,accion,fecha)
  VALUES ('Antonio Pardo','Borrar proveedor',now());
END
//
#Se llama el procedimiento creado para borrar proveedor
CALL borrar_proveedor(1);

```

En este trigger se permite registrar el momento en que se elimina un proveedor y quien realizo esta acción

3. TRIGGER registro_agregar_proveedor

```
# 3. CREACION DE TRIGGER PARA REGISTRAR ACCION DE CREACION DE PROVEEDOR
DELIMITER //
CREATE TRIGGER registro_agregar_proveedor
AFTER INSERT
ON proveedor FOR EACH ROW
BEGIN
INSERT INTO control_de_cambios_zoologico (usuario,accion,fecha)
VALUES ('Miguel Molina','Agregar proveedor',now());
END
//
#Se usa el procedimiento agregar proveedor para cargar el registro
CALL agregarProveedor('12','Aves','Cali','Aves@agro.com');

#se lista la tablacontrol_de_cambios_libreria
SELECT * FROM control_de_cambios_zoologico;
```

Se permite registrar el momento en que agrega un proveedor, de igual manera quien realiza esta acción. Esta información es cargada en la tabla control_de_cambios_zoologico.

4. TRIGGER registro_agregar_alimento

```
# 4. CREACION DE TRIGGER PARA REGISTRAR ACCION DE CREACION DE ALIMENTO
DELIMITER //
CREATE TRIGGER registro_agregar_alimento
AFTER INSERT
ON alimento FOR EACH ROW
BEGIN
INSERT INTO control_de_cambios_zoologico (usuario,accion,fecha)
VALUES ('Angel Mora','Agregar alimento',now());
END
//
#Se agrega el alimento
INSERT INTO alimento (id_alimento,nombre,dosis) VALUES ('A10','Purina pez','50 gramos');

#se lista la tablacontrol_de_cambios_libreria
SELECT * FROM control_de_cambios_zoologico;
```

En este trigger permite al realizar el registro de un nuevo alimento en la base de datos conocer quien realizo esta acción, la fecha y hora de estos registros.

SE INGRESA INFORMACION A CADA TABLA:

Por último, se procede a cargar la base de registros en cada una de sus tablas conectando la base de datos desde Java, **se cargan 50 registros para cada una.**

Se crearon las clases de cada tabla donde de especifica cada atributo, método constructor y métodos setters y getters, se crean interfaces que permiten plantear las acciones a realizar y cada clase donde se implementa lo estipulado en cada interfaz. En la clase Main se cargan los registros para cada tabla por medio de la librería **Faker** que permite crear registros para cargar las tablas de manera que se comprueba su funcionamiento.

TABLA VETERINARIO:

```
> task :classes
> Task :Main.main()
ID: PR-0  NOMBRE: Leeann  TELEFONO: (972) 701-6045
ID: PR-1  NOMBRE: Angelia  TELEFONO: 670.549.7187
ID: PR-10 NOMBRE: Cristin  TELEFONO: 579-149-8460
ID: PR-11 NOMBRE: Lacey  TELEFONO: 1-720-752-1557
ID: PR-12 NOMBRE: Usha  TELEFONO: (417) 363-4813
ID: PR-13 NOMBRE: Karry  TELEFONO: 098-259-3571
ID: PR-14 NOMBRE: Shelly  TELEFONO: (821) 633-4592
ID: PR-15 NOMBRE: Ashely  TELEFONO: (357) 320-8583
ID: PR-16 NOMBRE: Su  TELEFONO: 400-631-5187
ID: PR-17 NOMBRE: Fabian  TELEFONO: (413) 214-3301
ID: PR-18 NOMBRE: Mikki  TELEFONO: 047-474-4355
ID: PR-19 NOMBRE: Johnette  TELEFONO: (506) 090-4957
ID: PR-2  NOMBRE: Lourie  TELEFONO: 714-351-0185
ID: PR-20 NOMBRE: Felipe  TELEFONO: 1-432-473-3837
ID: PR-21 NOMBRE: Abraham  TELEFONO: 298-279-7436
ID: PR-22 NOMBRE: Jeromy  TELEFONO: 321.656.7164
ID: PR-23 NOMBRE: Paris  TELEFONO: 581.536.5612
ID: PR-24 NOMBRE: Bernie  TELEFONO: 1-832-136-1051
ID: PR-25 NOMBRE: Carli  TELEFONO: 1-413-037-2452
ID: PR-26 NOMBRE: Luanna  TELEFONO: 904.110.7239
ID: PR-27 NOMBRE: Davis  TELEFONO: 051.416.7610
ID: PR-28 NOMBRE: Ranae  TELEFONO: 1-243-792-3183
ID: PR-31 NOMBRE: Wally  TELEFONO: 1-893-434-3014
ID: PR-32 NOMBRE: Maida  TELEFONO: 376.632.6989
ID: PR-33 NOMBRE: Jenny  TELEFONO: (146) 664-4662
ID: PR-34 NOMBRE: Lashawna  TELEFONO: 1-491-749-1497
ID: PR-35 NOMBRE: Enriqueta  TELEFONO: 979.059.9478
ID: PR-36 NOMBRE: Nicole  TELEFONO: 355-392-5852
ID: PR-37 NOMBRE: Jarvis  TELEFONO: 142-766-8300
ID: PR-38 NOMBRE: Esther  TELEFONO: 151-982-9374
ID: PR-39 NOMBRE: Vern  TELEFONO: 565-259-9154
ID: PR-4  NOMBRE: Jeanett  TELEFONO: (691) 617-4435
ID: PR-40 NOMBRE: Kimberly  TELEFONO: 1-461-675-5546
ID: PR-41 NOMBRE: Olga  TELEFONO: 1-997-093-3818
ID: PR-42 NOMBRE: Rich  TELEFONO: 866.092.1781
ID: PR-43 NOMBRE: Layla  TELEFONO: (018) 130-0535
ID: PR-44 NOMBRE: Vickey  TELEFONO: 059.435.6781
ID: PR-45 NOMBRE: Hobert  TELEFONO: 1-151-829-0271
ID: PR-46 NOMBRE: Yukiko  TELEFONO: 971-470-4869
ID: PR-47 NOMBRE: Stacie  TELEFONO: (222) 213-2766
ID: PR-48 NOMBRE: Jess  TELEFONO: 230.104.9544
ID: PR-49 NOMBRE: Cruz  TELEFONO: 1-394-778-3324
ID: PR-5  NOMBRE: Clint  TELEFONO: 1-641-138-5793
ID: PR-50 NOMBRE: Benny  TELEFONO: (182) 874-6472
ID: PR-6  NOMBRE: Arturo  TELEFONO: 666-476-1848
ID: PR-7  NOMBRE: Tiera  TELEFONO: 197-422-6588
ID: PR-8  NOMBRE: Charlsie  TELEFONO: (909) 561-6833
ID: PR-9  NOMBRE: Joeann  TELEFONO: 343-642-1262
```

Esto son los datos cargados en java, se confirma información desde la base de datos obteniendo:

id_veterinario	nombre	telefono	id_veterinario	nombre	telefono
PR-0	Leeann	(972) 701-6045	PR-44	Vickey	059.435.6781
PR-1	Angelia	670.549.7187	PR-45	Hobert	1-151-829-0271
PR-10	Cristin	579-149-8460	PR-46	Yukiko	971-470-4869
PR-11	Lacey	1-720-752-1557	PR-47	Stacie	(222) 213-2766
PR-12	Usha	(417) 363-4813	PR-48	Jess	230.104.9544
PR-13	Karry	098-259-3571	PR-49	Cruz	1-394-778-3324
PR-14	Shelly	(821) 633-4592	PR-5	Clint	1-641-138-5793
PR-15	Ashely	(357) 320-8583	PR-50	Benny	(182) 874-6472
PR-16	Su	400-631-5187	PR-6	Arturo	666-476-1848
PR-17	Fabian	(413) 214-3301	PR-7	Tiera	197-422-6588
PR-18	Mikki	047-474-4355	PR-8	Charlsie	(909) 561-6833
PR-19	Johnette	(506) 090-4957	PR-9	Joeann	343-642-1262

Se puede ver que los registros coinciden a plenitud.

TABLA PROVEEDOR:

ID: PRO-0	NOMBRE: Tegan	DIRECCION: Rowebury	CORREO: tyler.sawayn@gmail.com
ID: PRO-1	NOMBRE: Millie	DIRECCION: Paucekfurt	CORREO: edyth.nader@hotmail.com
ID: PRO-10	NOMBRE: Norris	DIRECCION: Mayerview	CORREO: sheryl.sawayn@yahoo.com
ID: PRO-11	NOMBRE: Melinda	DIRECCION: East Lemuel	CORREO: denver.bergnaum@yahoo.com
ID: PRO-12	NOMBRE: Dahlia	DIRECCION: New Lupe	CORREO: emelia.armstrong@hotmail.com
ID: PRO-13	NOMBRE: Kristin	DIRECCION: Mohrville	CORREO: jeri.ward@yahoo.com
ID: PRO-14	NOMBRE: Alta	DIRECCION: East Shaquitastad	CORREO: alexander.braven@gmail.com
ID: PRO-15	NOMBRE: Bobby	DIRECCION: Burmabury	CORREO: joana.olson@gmail.com
ID: PRO-16	NOMBRE: September	DIRECCION: Bauchbury	CORREO: otto.skiles@yahoo.com
ID: PRO-17	NOMBRE: Lissa	DIRECCION: O'Reillymouth	CORREO: darell.maggio@gmail.com
ID: PRO-18	NOMBRE: Matha	DIRECCION: Lake Zackton	CORREO: mac.lockman@yahoo.com
ID: PRO-19	NOMBRE: Kerry	DIRECCION: North Zackville	CORREO: demetrice.okuneva@yahoo.com
ID: PRO-2	NOMBRE: Yvone	DIRECCION: Schulisttown	CORREO: carlo.anderson@gmail.com
ID: PRO-20	NOMBRE: Jerrold	DIRECCION: Deloisietown	CORREO: todd.schulist@hotmail.com
ID: PRO-21	NOMBRE: Nathan	DIRECCION: North Jimview	CORREO: otis.sanford@gmail.com
ID: PRO-22	NOMBRE: Marleen	DIRECCION: West Hilario	CORREO: ira.bins@gmail.com
ID: PRO-23	NOMBRE: Rosamaria	DIRECCION: East Jospeh	CORREO: maryann.ferry@hotmail.com
ID: PRO-24	NOMBRE: Laurel	DIRECCION: Diedraburgh	CORREO: ivory.donnelly@hotmail.com
ID: PRO-25	NOMBRE: Tod	DIRECCION: East Margrettown	CORREO: ricarda.heaney@yahoo.com
ID: PRO-26	NOMBRE: Jeremy	DIRECCION: West Blaine	CORREO: giovanni.thiel@yahoo.com
ID: PRO-27	NOMBRE: Dovie	DIRECCION: Lindtown	CORREO: mohamed.weissnat@hotmail.com
ID: PRO-28	NOMBRE: Agustina	DIRECCION: North Royal	CORREO: gayla.rodriguez@yahoo.com
ID: PRO-29	NOMBRE: Milagros	DIRECCION: West Sue	CORREO: arthur.bartell@gmail.com
ID: PRO-3	NOMBRE: Candyce	DIRECCION: Mollybury	CORREO: randolph.crooks@yahoo.com
ID: PRO-30	NOMBRE: Maria	DIRECCION: Fayberg	CORREO: danyell.johnston@yahoo.com

Estos son los datos cargados desde java y posteriormente consultados desde MySQL

id_proveedor	nombre	direccion	correo
PRO-30	Maria	Fayberg	danyell.johnston@yahoo.com
PRO-31	Brunilda	Kelsiefurt	nolan.vandervort@gmail.com
PRO-32	Donita	Krissyshire	celena.west@gmail.com
PRO-33	Faye	New Trentborough	glinda.casper@hotmail.com
PRO-34	Dallas	New Dana	rosana.denesik@yahoo.com
PRO-35	Cyril	Schmittshire	sebastian.stehr@gmail.com
PRO-36	Hans	Raphaelburgh	valentin.white@gmail.com
PRO-37	Dewitt	Sammyville	miguel.bahringer@yahoo.com
PRO-38	Suzi	Hansburgh	man.hartmann@yahoo.com
PRO-39	Kandace	South Mitchellshire	deshawn.nolan@hotmail.com
PRO-4	Dane	Arnoldoburgh	lucius.weissnat@yahoo.com
PRO-40	Mel	Illaside	wendell.tremblay@yahoo.com
PRO-41	Herman	Hirtheaton	tommie.reilly@yahoo.com
PRO-42	Taylor	South Alphonse	ward.mitchell@hotmail.com
PRO-43	Darin	South Roosevelt	sandra.wunsch@hotmail.com
PRO-44	Ashley	Borerberg	simonne.walker@hotmail.com
PRO-45	Nia	Bradfordburgh	martin.cole@gmail.com
PRO-46	Santos	Kevenshire	gabriella.fadel@yahoo.com
PRO-47	Carmelina	Lake Claudio	alvin.baumbach@gmail.com

proveedor 14 v

TABLA ALIMENTO:

ID	NOMBRE	DOSIS
ASR-32	Corn Tortilla	5
ASR-33	Galangal	8
ASR-34	Loquats	8
ASR-35	Amaranth	0
ASR-36	Camembert	4
ASR-37	Swordfish	8
ASR-38	Mahlab	6
ASR-39	Calamari	7
ASR-4	Mung Beans	9
ASR-40	Peas	8
ASR-41	Kidneys	6
ASR-42	Lavender Flowers	0
ASR-43	Buttermilk	4
ASR-44	Mahlab	3
ASR-45	Cloves	5
ASR-46	Dandelion	5
ASR-47	Tangelo	5
ASR-48	Whiting Wild Rice	4
ASR-49	Kangaroo	2
ASR-5	Custard Apples	3
ASR-50	Yellow Papaw	4
ASR-6	Kiwi Fruit	5
ASR-7	Tangelo	4
ASR-8	Swiss Chard	8
ASR-9	Chicken Stock	6

id_alimento	nombre	dosis
ASR-16	Zucchini	5
ASR-17	Buttermilk	6
ASR-18	Mahlab	3
ASR-19	Mozzarella	6
ASR-2	Ajowan Seed	3
ASR-20	Raspberry	1
ASR-21	Chinese Five Spice	7
ASR-22	Plums	8
ASR-23	Fish Stock	5
ASR-24	Nutmeg	8
ASR-25	Hokkien Noodles	7
ASR-26	Cayenne	7
ASR-27	Kiwi Berries	0
ASR-28	Wholewheat Flour	5
ASR-29	Coriander Leaves	7
ASR-3	Tinned	8
ASR-30	Oats	5
ASR-31	Beetroot	7
ASR-32	Corn Tortilla	5

Se cargan datos a la tabla desde Java y se consultan posteriormente en MySQL, se activa trigger al ingresar un nuevo alimento se visualiza usuario, acción realizada y fecha y hora del registro.

usuario	accion	fecha
Angel Mora	Agregar alimento	2023-02-19 00:43:58
Angel Mora	Agregar alimento	2023-02-19 00:44:00
Angel Mora	Agregar alimento	2023-02-19 00:44:01
Angel Mora	Agregar alimento	2023-02-19 00:44:01
Angel Mora	Agregar alimento	2023-02-19 00:44:02
Angel Mora	Agregar alimento	2023-02-19 00:44:02
Angel Mora	Agregar alimento	2023-02-19 00:44:03
Angel Mora	Agregar alimento	2023-02-19 00:44:03
Angel Mora	Agregar alimento	2023-02-19 00:44:03
Angel Mora	Agregar alimento	2023-02-19 00:44:04
Angel Mora	Agregar alimento	2023-02-19 00:44:06
Angel Mora	Agregar alimento	2023-02-19 00:44:06
Angel Mora	Agregar alimento	2023-02-19 00:44:06
Angel Mora	Agregar alimento	2023-02-19 00:44:06

TABLA ANIMAL

```
ID: ANIM-0  NOMBRE: weasel  SEXO: male  FECHA NACIMIENTO: senior  EDAD: young
ID: ANIM-1  NOMBRE: giraffe  SEXO: female  FECHA NACIMIENTO: puppy  EDAD: puppy
ID: ANIM-10  NOMBRE: turtle  SEXO: female  FECHA NACIMIENTO: senior  EDAD: senior
ID: ANIM-11  NOMBRE: badger  SEXO: female  FECHA NACIMIENTO: senior  EDAD: puppy
ID: ANIM-12  NOMBRE: ferret  SEXO: male  FECHA NACIMIENTO: young  EDAD: young
ID: ANIM-13  NOMBRE: wasp  SEXO: female  FECHA NACIMIENTO: adult  EDAD: young
ID: ANIM-14  NOMBRE: ape  SEXO: female  FECHA NACIMIENTO: senior  EDAD: adult
ID: ANIM-15  NOMBRE: raven  SEXO: male  FECHA NACIMIENTO: young  EDAD: young
ID: ANIM-16  NOMBRE: salmon  SEXO: female  FECHA NACIMIENTO: adult  EDAD: puppy
ID: ANIM-17  NOMBRE: swan  SEXO: male  FECHA NACIMIENTO: young  EDAD: young
ID: ANIM-18  NOMBRE: tiger  SEXO: female  FECHA NACIMIENTO: adult  EDAD: puppy
ID: ANIM-19  NOMBRE: yellowjacket  SEXO: female  FECHA NACIMIENTO: puppy  EDAD: senior
ID: ANIM-2  NOMBRE: fly  SEXO: male  FECHA NACIMIENTO: puppy  EDAD: puppy
ID: ANIM-20  NOMBRE: raven  SEXO: male  FECHA NACIMIENTO: adult  EDAD: adult
ID: ANIM-21  NOMBRE: rat  SEXO: male  FECHA NACIMIENTO: adult  EDAD: adult
ID: ANIM-22  NOMBRE: zebra  SEXO: male  FECHA NACIMIENTO: senior  EDAD: adult
ID: ANIM-23  NOMBRE: dinosaur  SEXO: male  FECHA NACIMIENTO: senior  EDAD: adult
ID: ANIM-24  NOMBRE: snake  SEXO: female  FECHA NACIMIENTO: adult  EDAD: adult
ID: ANIM-25  NOMBRE: tortoise  SEXO: male  FECHA NACIMIENTO: adult  EDAD: puppy
ID: ANIM-26  NOMBRE: scorpion  SEXO: male  FECHA NACIMIENTO: adult  EDAD: young
ID: ANIM-27  NOMBRE: frog  SEXO: male  FECHA NACIMIENTO: young  EDAD: puppy
ID: ANIM-28  NOMBRE: gnat  SEXO: male  FECHA NACIMIENTO: adult  EDAD: puppy
ID: ANIM-29  NOMBRE: raven  SEXO: male  FECHA NACIMIENTO: puppy  EDAD: puppy
ID: ANIM-3  NOMBRE: ox  SEXO: male  FECHA NACIMIENTO: senior  EDAD: adult
ID: ANIM-30  NOMBRE: seal  SEXO: female  FECHA NACIMIENTO: senior  EDAD: senior
```

id_animal	nombre	sexo	fecha_nacimiento	edad
ANIM-38	gnu	female	puppy	young
ANIM-39	goldfish	male	adult	puppy
ANIM-4	ape	male	adult	adult
ANIM-40	elk	male	senior	puppy
ANIM-41	giraffe	male	puppy	senior
ANIM-42	impala	female	young	young
ANIM-43	platypus	female	puppy	puppy
ANIM-44	spider	male	puppy	adult
ANIM-45	raven	female	senior	senior
ANIM-46	goldfish	male	adult	puppy
ANIM-47	mallard	male	adult	young
ANIM-48	gnu	female	adult	puppy
ANIM-49	kangaroo	male	adult	young
ANIM-5	raven	female	young	young
ANIM-50	chinchilla	male	young	young
ANIM-6	elephant	male	adult	young
ANIM-7	tortoise	male	senior	adult
ANIM-8	hyena	male	young	puppy
ANIM-9	horse	male	puppy	senior

Se registran los datos de la tabla animal desde Java y se consultan posteriormente desde MySQL.

TABLA ENTRENADOR

ID: ENTREN-0	NOMBRE: Sheryl	TELEFONO: (357) 946-3549	VETERINARIO A REPORTAR PESO: PR-0
ID: ENTREN-1	NOMBRE: Darrick	TELEFONO: 323.238.3872	VETERINARIO A REPORTAR PESO: PR-1
ID: ENTREN-10	NOMBRE: Bernard	TELEFONO: (532) 658-9964	VETERINARIO A REPORTAR PESO: PR-10
ID: ENTREN-11	NOMBRE: Manuel	TELEFONO: 1-216-479-0311	VETERINARIO A REPORTAR PESO: PR-11
ID: ENTREN-12	NOMBRE: Zane	TELEFONO: 1-623-557-7667	VETERINARIO A REPORTAR PESO: PR-12
ID: ENTREN-13	NOMBRE: Sadye	TELEFONO: 1-230-219-3239	VETERINARIO A REPORTAR PESO: PR-13
ID: ENTREN-14	NOMBRE: Rima	TELEFONO: 084.014.4806	VETERINARIO A REPORTAR PESO: PR-14
ID: ENTREN-15	NOMBRE: Thurman	TELEFONO: 091.826.7401	VETERINARIO A REPORTAR PESO: PR-15
ID: ENTREN-16	NOMBRE: Zoe	TELEFONO: 870.167.3024	VETERINARIO A REPORTAR PESO: PR-16
ID: ENTREN-17	NOMBRE: Neoma	TELEFONO: 182.346.1553	VETERINARIO A REPORTAR PESO: PR-17
ID: ENTREN-18	NOMBRE: Reyes	TELEFONO: 1-612-514-0735	VETERINARIO A REPORTAR PESO: PR-18
ID: ENTREN-19	NOMBRE: Junie	TELEFONO: 336.034.4977	VETERINARIO A REPORTAR PESO: PR-19
ID: ENTREN-2	NOMBRE: Lenna	TELEFONO: 1-228-092-9848	VETERINARIO A REPORTAR PESO: PR-2
ID: ENTREN-20	NOMBRE: Henry	TELEFONO: 429.612.7393	VETERINARIO A REPORTAR PESO: PR-20
ID: ENTREN-21	NOMBRE: Florencio	TELEFONO: (905) 456-6996	VETERINARIO A REPORTAR PESO: PR-21
ID: ENTREN-22	NOMBRE: Gonzalo	TELEFONO: 519.829.3438	VETERINARIO A REPORTAR PESO: PR-22
ID: ENTREN-23	NOMBRE: Ira	TELEFONO: 1-950-103-1511	VETERINARIO A REPORTAR PESO: PR-23
ID: ENTREN-24	NOMBRE: Benito	TELEFONO: 658.774.6692	VETERINARIO A REPORTAR PESO: PR-24
ID: ENTREN-25	NOMBRE: AnnaLee	TELEFONO: 596-501-0905	VETERINARIO A REPORTAR PESO: PR-25
ID: ENTREN-26	NOMBRE: Graig	TELEFONO: (751) 542-7571	VETERINARIO A REPORTAR PESO: PR-26
ID: ENTREN-27	NOMBRE: Alyssa	TELEFONO: 230-599-6192	VETERINARIO A REPORTAR PESO: PR-27
ID: ENTREN-28	NOMBRE: Chang	TELEFONO: (561) 043-2019	VETERINARIO A REPORTAR PESO: PR-28
ID: ENTREN-29	NOMBRE: Denese	TELEFONO: (226) 204-3785	VETERINARIO A REPORTAR PESO: PR-29
ID: ENTREN-3	NOMBRE: Isiah	TELEFONO: 1-093-824-0997	VETERINARIO A REPORTAR PESO: PR-3
ID: ENTREN-30	NOMBRE: Adolph	TELEFONO: (595) 883-3621	VETERINARIO A REPORTAR PESO: PR-30
ID: ENTREN-31	NOMBRE: Elinor	TELEFONO: 1-695-540-8878	VETERINARIO A REPORTAR PESO: PR-31

Se cargan los registros en la tabla entrenador desde Java y se consultan posteriormente desde MySQL:

id_entrenador	nombre	telefono	id_veterinarios
ENTREN-38	Sid	368-911-9927	PR-38
ENTREN-39	Terence	472-845-4151	PR-39
ENTREN-4	Homer	1-647-631-9944	PR-4
ENTREN-40	Kristine	1-661-655-2930	PR-40
ENTREN-41	Jospeh	1-195-542-8758	PR-41
ENTREN-42	Delphine	933.517.2134	PR-42
ENTREN-43	Aaron	(259) 579-4976	PR-43
ENTREN-44	Galina	(763) 752-2685	PR-44
ENTREN-45	Karri	(661) 498-5640	PR-45
ENTREN-46	Demetria	477.231.7111	PR-46
ENTREN-47	Paul	1-766-593-3951	PR-47
ENTREN-48	Antione	1-229-431-5552	PR-48
ENTREN-49	Otto	009.496.3734	PR-49
ENTREN-5	Prince	(348) 154-2583	PR-5
ENTREN-50	Alan	911-966-0101	PR-50
ENTREN-6	Kortney	1-248-892-4693	PR-6
ENTREN-7	Tyrone	1-138-909-1114	PR-7
ENTREN-8	Dani	1-455-521-6832	PR-8
ENTREN-9	Odell	414.566.6492	PR-9

TABLA EMPLEADO LOGISTICO

id_logistico	nombre	correo
ELOG-0	Shanti	gil.schiller@gmail.com
ELOG-1	Yael	denita.hegmann@hotmail.com
ELOG-10	Ellie	boyce.ohara@yahoo.com
ELOG-11	Shin	maurice.schuppe@yahoo.com
ELOG-12	Vi	derek.schmeler@yahoo.com
ELOG-13	Marisol	henry.mayert@gmail.com
ELOG-14	Selina	kenyatta.anderson@hotmail.com
ELOG-15	Gilberto	rosalee.trantow@hotmail.com
ELOG-16	Lawanda	alene.ratke@gmail.com
ELOG-17	Donte	dorthey.mertz@hotmail.com
ELOG-18	Laree	fe.fadel@hotmail.com
ELOG-19	Maxima	roland.turner@gmail.com
ELOG-2	Rohin	ronnv.dickens@gmail.com

Se muestran los datos cargados en la tabla empleado logístico, estos registros fueron cargados desde Java y se muestran los resultados de esta acción por medio de una consulta en MySQL.

TABLA DIETA

```
ID: DIE-0 RECOMENDACIONES: Provident magnam dolore expedita. ANIMAL A ASIGNAR DIETA: ANIM-0
ID: DIE-1 RECOMENDACIONES: Nobis consectetur dolor neque est quo. ANIMAL A ASIGNAR DIETA: ANIM-1
ID: DIE-12 RECOMENDACIONES: Debitis magnam recusandae ut. ANIMAL A ASIGNAR DIETA: ANIM-12
ID: DIE-14 RECOMENDACIONES: Quo et cumque ut et. ANIMAL A ASIGNAR DIETA: ANIM-14
ID: DIE-17 RECOMENDACIONES: Non ullam ea ex qui blanditiis. ANIMAL A ASIGNAR DIETA: ANIM-17
ID: DIE-18 RECOMENDACIONES: Quia et consequatur. ANIMAL A ASIGNAR DIETA: ANIM-18
ID: DIE-19 RECOMENDACIONES: Modi repellendus nesciunt. ANIMAL A ASIGNAR DIETA: ANIM-19
ID: DIE-2 RECOMENDACIONES: Omnis ipsum ut eligendi et quia. ANIMAL A ASIGNAR DIETA: ANIM-2
ID: DIE-25 RECOMENDACIONES: Amet ipsa voluptatem. ANIMAL A ASIGNAR DIETA: ANIM-25
ID: DIE-27 RECOMENDACIONES: Rem sit quaerat harum non laudantium non. ANIMAL A ASIGNAR DIETA: ANIM-27
ID: DIE-28 RECOMENDACIONES: Qui autem deserunt quia. ANIMAL A ASIGNAR DIETA: ANIM-28
ID: DIE-29 RECOMENDACIONES: Ducimus et totam voluptatem asperiores quasi. ANIMAL A ASIGNAR DIETA: ANIM-29
ID: DIE-3 RECOMENDACIONES: Deleniti sit nesciunt. ANIMAL A ASIGNAR DIETA: ANIM-3
ID: DIE-30 RECOMENDACIONES: Optio est dolore et beatae. ANIMAL A ASIGNAR DIETA: ANIM-30
ID: DIE-31 RECOMENDACIONES: Natus eius numquam est ab. ANIMAL A ASIGNAR DIETA: ANIM-31
ID: DIE-34 RECOMENDACIONES: Ut debitis fuga nisi non quis. ANIMAL A ASIGNAR DIETA: ANIM-34
ID: DIE-36 RECOMENDACIONES: Qui est quisquam rem. ANIMAL A ASIGNAR DIETA: ANIM-36
ID: DIE-39 RECOMENDACIONES: Doloribus ut eos. ANIMAL A ASIGNAR DIETA: ANIM-39
ID: DIE-4 RECOMENDACIONES: Id aut natus possimus. ANIMAL A ASIGNAR DIETA: ANIM-4
ID: DIE-40 RECOMENDACIONES: Facere quia numquam facilis iure occaecati. ANIMAL A ASIGNAR DIETA: ANIM-40
ID: DIE-41 RECOMENDACIONES: Et et et voluptate ipsum voluptatibus et. ANIMAL A ASIGNAR DIETA: ANIM-41
ID: DIE-43 RECOMENDACIONES: Perferendis reprehenderit facilis. ANIMAL A ASIGNAR DIETA: ANIM-43
ID: DIE-44 RECOMENDACIONES: Perferendis minus quia aliquam. ANIMAL A ASIGNAR DIETA: ANIM-44
ID: DIE-45 RECOMENDACIONES: Repudiandae incidunt minus. ANIMAL A ASIGNAR DIETA: ANIM-45
ID: DIE-46 RECOMENDACIONES: Qui qui dolore error soluta laboriosam. ANIMAL A ASIGNAR DIETA: ANIM-46
```

id_dieta	recomendaciones	id_animals
DIE-4	Id aut natus possimus.	ANIM-4
DIE-40	Facere quia numquam facilis iure occae...	ANIM-40
DIE-41	Et et et voluptate ipsum voluptatibus et.	ANIM-41
DIE-43	Perferendis reprehenderit facilis.	ANIM-43
DIE-44	Perferendis minus quia aliquam.	ANIM-44
DIE-45	Repudiandae incidunt minus.	ANIM-45
DIE-46	Qui qui dolore error soluta laboriosam.	ANIM-46
DIE-47	Dolorum non iusto doloreque.	ANIM-47
DIE-5	Expedita veritatis saepe impedit libero ...	ANIM-5
DIE-50	Dicta voluptatem quos.	ANIM-50
DIE-7	Quis id facere.	ANIM-7
DIE-8	Et eaque at voluptatibus aut quia natus.	ANIM-8
DIE-9	Quibusdam reiciendis qui dolor repellend...	ANIM-9

Se cargan datos a la tabla dieta desde Java, se consulta resultado desde MySQL.

TABLA FACTURA

ID: FAC-0	FECHA: 213-24-6575	TOTAL 004-67-2169
ID: FAC-1	FECHA: 597-72-7619	TOTAL 619-11-6174
ID: FAC-10	FECHA: 195-42-7292	TOTAL 233-97-8893
ID: FAC-11	FECHA: 582-25-0228	TOTAL 735-16-4702
ID: FAC-12	FECHA: 510-74-5981	TOTAL 059-13-6994
ID: FAC-13	FECHA: 207-82-8247	TOTAL 261-50-9978
ID: FAC-14	FECHA: 674-84-9623	TOTAL 885-82-6576
ID: FAC-15	FECHA: 546-80-3269	TOTAL 755-40-5223
ID: FAC-16	FECHA: 425-64-6061	TOTAL 201-45-2003
ID: FAC-17	FECHA: 536-86-9483	TOTAL 134-13-4170
ID: FAC-18	FECHA: 248-91-8793	TOTAL 268-28-5612
ID: FAC-19	FECHA: 684-14-0711	TOTAL 609-12-8555
ID: FAC-2	FECHA: 262-74-6266	TOTAL 658-52-2560
ID: FAC-20	FECHA: 259-84-2729	TOTAL 058-82-1112
ID: FAC-21	FECHA: 225-27-3798	TOTAL 301-82-5552
ID: FAC-22	FECHA: 762-53-0897	TOTAL 649-63-7651
ID: FAC-23	FECHA: 015-73-6233	TOTAL 143-66-6175
ID: FAC-24	FECHA: 419-94-1878	TOTAL 724-20-5537
ID: FAC-25	FECHA: 046-86-2140	TOTAL 017-57-8251
ID: FAC-26	FECHA: 226-68-0507	TOTAL 789-55-7802
ID: FAC-27	FECHA: 584-66-8791	TOTAL 791-27-4290
ID: FAC-28	FECHA: 620-33-0768	TOTAL 454-03-1006
ID: FAC-29	FECHA: 147-44-7908	TOTAL 158-80-8408
ID: FAC-3	FECHA: 000-90-2067	TOTAL 100-66-8842
ID: FAC-30	FECHA: 510-67-9462	TOTAL 872-76-1483

Se cargan datos a la tabla factura desde Java y se consultan posteriormente desde MySQL.

id_factura	fecha	total
FAC-43	775-19-5332	746-65-2879
FAC-44	278-42-9218	669-22-0190
FAC-45	645-46-3690	139-88-7297
FAC-46	329-41-5380	541-91-0954
FAC-47	135-59-7821	156-89-1775
FAC-48	201-94-2604	715-59-8692
FAC-49	309-05-8639	243-38-2271
FAC-5	777-22-7457	350-38-2267
FAC-50	739-01-7420	540-30-1932
FAC-6	724-90-8466	424-63-3466
FAC-7	300-01-6764	845-51-6784
FAC-8	070-01-0717	022-99-5988
FAC-9	764-29-6775	878-88-4849

TABLA ORDEN DE COMPRA

id_orden	fecha	nombre_alimento	cantidad	id_logisticos	id_proveedores	observaciones
OR-0	274-99-0996	Raspberry	5	ELOG-0	PRO-0	Impedit ut vel qui repellat ab nobis modi.
OR-1	014-05-3301	Edamame	3	ELOG-1	PRO-1	Eligendi voluptas occaecati non.
OR-10	772-53-0190	Gruyere	7	ELOG-10	PRO-10	Dicta quod asperiores possimus.
OR-11	448-53-6474	Macadamia Oil	0	ELOG-11	PRO-11	Eum ut ipsum maiores nulla voluptatem qui.
OR-12	495-69-8985	Dashi	0	ELOG-12	PRO-12	Dolores sint non qui.
OR-13	482-58-4843	Tom Yum	5	ELOG-13	PRO-13	Libero nesciunt ut omnis aut quod amet eos.
OR-14	206-77-6338	Celery	6	ELOG-14	PRO-14	Dolorum ullam sit cumque molestias.
OR-15	007-91-7024	Lychees	7	ELOG-15	PRO-15	Quam eos aliquid consequatur.
OR-16	346-84-3167	Blood Oranges	5	ELOG-16	PRO-16	Deserunt temporibus quia sit.
OR-17	670-38-2846	Pears	0	ELOG-17	PRO-17	Vero quae voluptatem debitis.
OR-18	108-43-1788	Tinned	3	ELOG-18	PRO-18	Et possimus aut incidunt repellat.
OR-19	179-81-8265	Apricots	3	ELOG-19	PRO-19	Ut quia esse ratione qui adipisci eos quas.
OR-21	865-21-7798	Rye Bread	8	ELOG-21	PRO-21	Sit similique nemo in totam.

Se cargan datos a la tabla orden de compra desde Java y se consulta posteriormente desde MySQL.

Esta misma acción se realiza con las tablas de las relaciones asociadas a cada una, se cargan 50 relaciones entre cada una.