

REPORTE DE PROYECTO FINAL

NOMBRES DE LOS ALUMNOS:

Ian Abishai Ramírez Olvera

Vladimir Islas Batalla

Joal David Sanchez Vidal

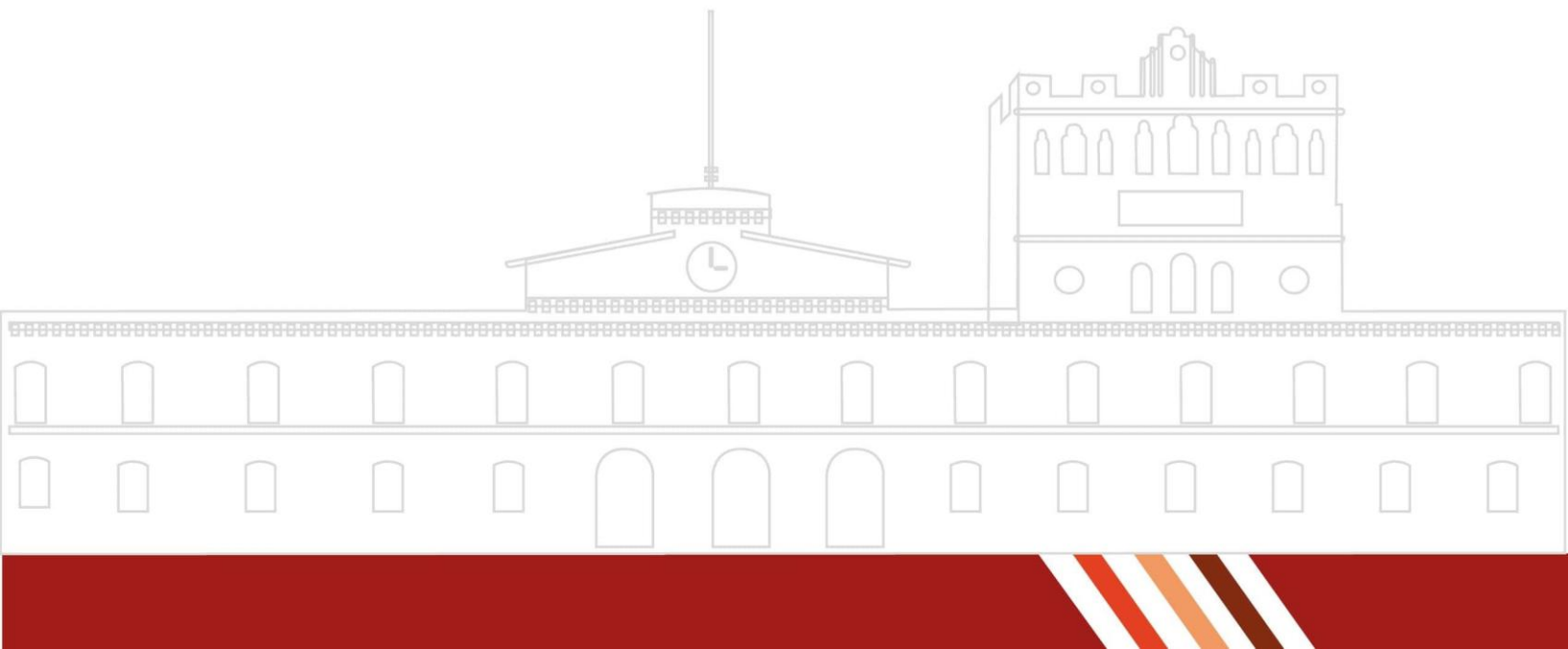
Yhojan Jesus Lozano Vazquez

Edwin Aaron Lopez Jimenez

Juan Pablo Ramos Hernadez

Dr. Eduardo Cornejo-Velázquez

Semestre y Grupo: 6° 2



1. Introducción

En la actualidad, el manejo eficiente de inventarios y ventas es un factor determinante para la popularidad para la productividad de las empresas dedicadas a la cría y comercialización de insectos, para consumo animal (principalmente reptiles). Sin embargo, muchas de estas organizaciones enfrentan problemas derivados de la falta de un sistema automatizado que permita llevar un control preciso sobre las especies, sus etapas de desarrollo, los lotes de producción, las fechas de compra y caducidad de los alimentos, así como las transacciones de venta y entrega a los clientes.

La gestión manual de estos procesos suele ocasionar errores en el registro de datos, pérdida de información, dificultades para rastrear el origen o el estado de los productos, y demoras en la toma de decisiones. Además, la ausencia de una base de datos estructurada limita la capacidad de generar reportes confiables y de mantener una trazabilidad adecuada de cada lote o pedido.

Ante esta problemática, se propone el desarrollo de una base de datos relacional que permita registrar, organizar y controlar la información relacionada con los usuarios del sistema, los alimentos utilizados, las especies de insectos producidas (grillos, cucarachas, zophobas y tenebrios), y las ventas realizadas. Este proyecto busca optimizar el control del inventario, mejorar la trazabilidad de los productos, reducir los errores de registro y fortalecer la gestión administrativa mediante una estructura de datos clara, normalizada y segura.

Desde una perspectiva operativa, el sistema permite optimizar mejor la producción, evitando tanto los excedentes como la escasez de insectos o alimentos. Al mantener registros históricos, se facilita la identificación de patrones de crecimiento, consumo o ventas, lo que abre la puerta a futuras mejoras en la logística y la gestión de recursos. Además, el registro de usuarios con diferentes rangos (administrador, gerente, operador, vendedor) refuerza la seguridad y control de accesos, garantizando que cada acción en el sistema esté asociada a un responsable.

Por otra parte, el impacto del sistema trasciende el nivel interno de la empresa, ya que mejora la confiabilidad de la información que se comparte con clientes o distribuidores. La posibilidad de ofrecer datos precisos sobre el estado de los pedidos, la disponibilidad de productos y los tiempos de entrega fortalece la relación con los clientes y eleva la competitividad del negocio. En conjunto, el desarrollo de esta base de datos constituye un paso fundamental hacia la transformación digital de la empresa, contribuyendo al control, eficiencia y sostenibilidad de sus operaciones.

2. Marco teórico

Análisis de requerimientos

En esta fase se identifica que información es necesario para cumplir con los objetivos del sistema. para el proyecto de gestión de insectos y ventas, los requerimientos son:

1. Usuarios del sistema: registrar información personal y credenciales de acceso.
2. Alimentos: registrar tipo, stock, fechas de compra y caducidad, y proveedor.
3. Insectos: registrar especies, etapa de desarrollo, stock, fechas de puesta y eclosión, y lotes de producción.
4. Ventas: registrar qué usuario realiza la venta, qué especie se vende, cantidad, etapa del insecto, fechas de salida y entrega, estado del pedido y cliente.
5. Seguridad: controlar el acceso de los usuarios según su rol, y mantener trazabilidad de las acciones realizadas en el sistema.

Identificación de entidades y atributos

A partir de los requerimientos, se identifican las siguientes entidades y sus atributos principales:

1. Usuarios: id usuario, nombre, apellido paterno, apellido materno, usuario, contraseña, rango, fecha registro.
2. Alimentos: id alimento, nombre, stock.kg, fecha compra, fecha caducidad, proveedor.
3. Insectos: id insecto, especie, stock, etapa, fecha puesta, fecha eclosion, id lote, fecha registro.
4. Ventas: id venta, id insecto, especie, cliente, cantidad, etapa, fecha salida, fecha entrega, estado, id usuario, fecha registro.

Roles de Usuario

Definen el nivel de acceso y permisos dentro del sistema:

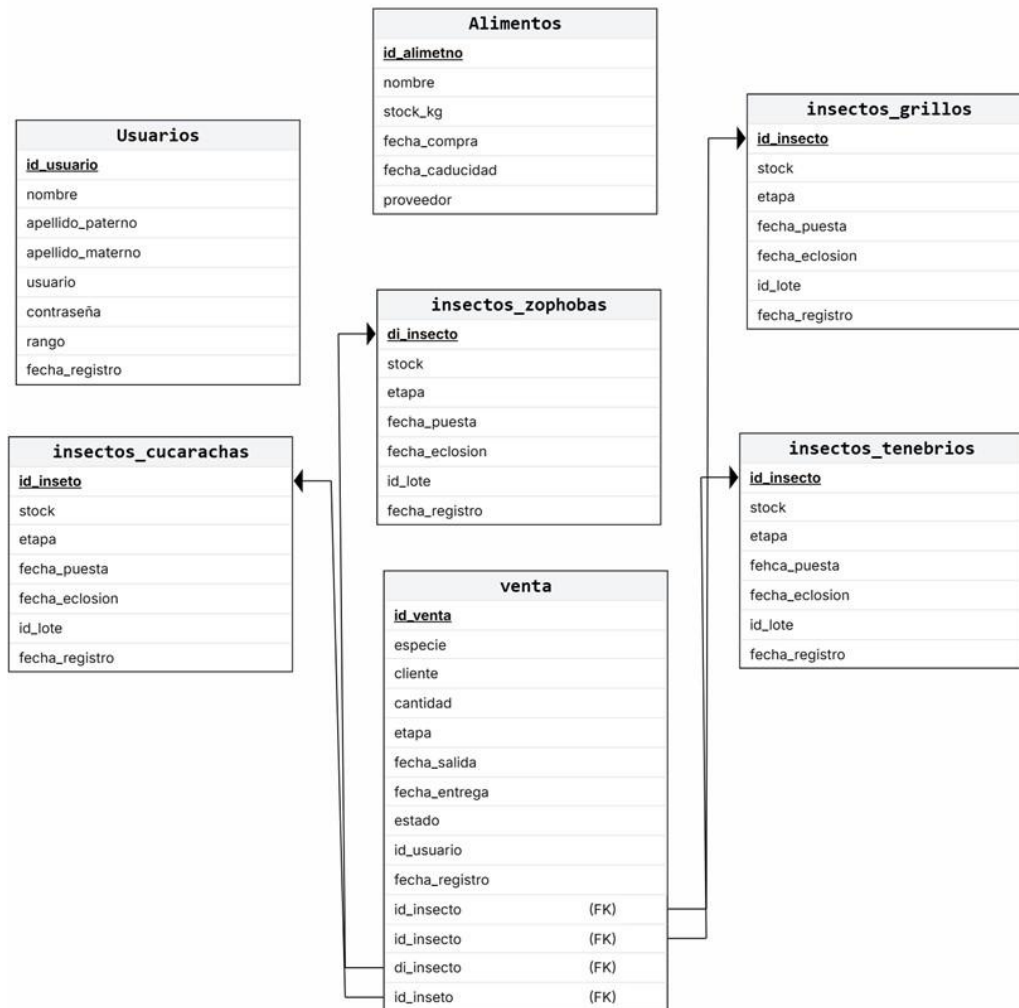
1. Administrador: Control total del sistema, gestión de usuarios, acceso a todos los módulos y reportes.
2. Gerente: Supervisión de inventarios y ventas, generación de reportes, acceso restringido a configuración de usuarios.
3. Operador: Manejo del inventario, registro de alimentos e insectos, actualización de stock.
4. Vendedor: Registro y seguimiento de ventas, generación de pedidos, acceso limitado a información de clientes y stock.

Roles de Seguridad

1. Control de acceso: cada usuario solo puede realizar acciones autorizadas según su rol.
2. Seguridad de la información: se protege información sensible como contraseñas y datos de clientes.
3. Trazabilidad: todas las acciones quedan registradas por usuario, facilitando auditorías y seguimiento de errores.
4. Prevención de errores: evita modificaciones no autorizadas en datos críticos, como stock o precios de venta.

Modelo relacional

El modelo relacional organiza los datos en tablas (o "relaciones") que constan de filas y columnas. Su característica principal es la capacidad de establecer vínculos entre las tablas mediante claves, lo que permite relacionar la información y consultarla de manera eficiente. Este modelo separa la estructura lógica de los datos de su almacenamiento físico, lo que facilita su gestión.



3. Desarrollo

Sentencias SQL

```
CREATE TABLE usuarios (  
  id_usuario INT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  apellido paterno VARCHAR(50) NOT NULL,  
  apellido materno VARCHAR(50),  
  telefono VARCHAR(20),  
  usuario VARCHAR(30) NOT NULL UNIQUE,  
  contraseña VARCHAR(255) NOT NULL,  
  rango ENUM('admin', 'gerente', 'operador', 'vendedor') DEFAULT 'operador',  
  fecha registro DATETIME  
);
```

```
CREATE TABLE alimentos (  
  id_alimento INT PRIMARY KEY,  
  nombre ENUM('calabaza', 'nopal', 'zanahoria') NOT NULL,  
  stock kg DECIMAL(6,2) NOT NULL DEFAULT 0.00,  
  fecha compra DATE NOT NULL,  
  fecha caducidad DATE,  
  proveedor VARCHAR(100),  
  CHECK (stock kg >= 0)  
);
```

```
CREATE TABLE insectos grillos(  
  id_insecto INT PRIMARY KEY,  
  stock INT NOT NULL DEFAULT 0,  
  etapa ENUM('huevo', 'larva', 'adulto') NOT NULL,  
  fecha puesta DATE  
  fecha eclosion DATE,  
  id_lote VARCHAR(20),  
  fecha_registro DATETIME,  
  CHECK (stock >= 0)  
);
```

```
CREATE TABLE insectos cucarachas(  
id_insecto INT PRIMARY KEY,  
stock INT NOT NULL DEFAULT 0,  
etapa ENUM('huevo', 'larva', 'adulto') NOT NULL,  
fecha_puesta DATE  
fecha_eclosion DATE,  
id_lote VARCHAR(20),  
fecha_registro DATETIME,  
CHECK (stock >= 0)  
);
```

```
CREATE TABLE insectos zophobas(  
id_insecto INT PRIMARY KEY,  
stock INT NOT NULL DEFAULT 0,  
etapa ENUM('huevo', 'larva', 'adulto') NOT NULL,  
fecha_puesta DATE  
fecha_eclosion DATE,  
id_lote VARCHAR(20),  
fecha_registro DATETIME,  
CHECK (stock >= 0)  
);
```

```
CREATE TABLE insectos tenebrios(  
id_insecto INT PRIMARY KEY,  
stock INT NOT NULL DEFAULT 0,  
etapa ENUM('huevo', 'larva', 'adulto') NOT NULL,  
fecha_puesta DATE  
fecha_eclosion DATE,  
id_lote VARCHAR(20),  
fecha_registro DATETIME,  
CHECK (stock >= 0)  
);
```

```

CREATE TABLE ventas(
id_ventas INT PRIMARY KEY,
id_insecto INT NOT NULL,
especie ENUM('grillo', 'cucaracha', 'zophobas', 'tenebrio') NOT NULL,
cliente ENUM('Petco') DEFAULT 'Petco',
etapa ENUM('huevo', 'ninfa', 'larva', 'adulto') NOT NULL,
cantidad INT NOT NULL,
fecha_salida DATE NOT NULL,
fecha_entrega DATE,
estado ENUM('pendiente', 'enviado', 'entregado', 'cancelado') DEFAULT
'pendiente',
id_usuario INT,
fecha_registro DATETIME DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),
fecha_registro DATETIME,
INDEX idx_fecha (fecha_salida),
INDEX idx_especie (especie),
CHECK (stock >= 0)
);

```

Insercion de datos(INSERT)

```

INSERT INTO usuarios (id_usuario, nombre, apellido_paterno, usuario, con-
traseña, rango, fecha_registro)
INSERT INTO usuarios (id_usuario, nombre, apellido_paterno, usuario, con-
traseña, rango, fecha_registro)
VALUES (1, 100, 'huevo', '2025-11-01', '2025-11-08', 'LoteG01', NOW());

```

Consultas (SELECT)

```

SELECT id_insecto, stock, etapa FROM insectos_grillos WHERE stock > 0;
SELECT * FROM ventas WHERE id_usuario = 1;

```

Actualización de datos (UPDATE)

```

UPDATE insectos_grillos
SET stock = stock - 10
WHERE id_insecto = 1;

```

Eliminación de datos (DELETE)

```

DELETE FROM ventas WHERE id_venta = 1;

```

Índices y optimización

```
CREATE INDEX idx fecha ON ventas(fecha_salida);  
CREATE INDEX idx especie ON ventas(especie);
```

4. Conclusiones

El desarrollo de esta base de datos permitió establecer una estructura sólida y organizada para la gestión de usuarios, alimentos, insectos y ventas dentro del sistema.

La implementación en SQL permitió definir de forma clara las relaciones entre las tablas, además de incorporar restricciones, índices y validaciones que mejoran la seguridad y el rendimiento. El uso de roles de usuario como administrador, gerente, operador y vendedor refuerza el control interno del sistema, asegurando que cada integrante tenga acceso únicamente a las funciones correspondientes a su nivel de responsabilidad.

Gracias a este diseño, la base de datos facilita la trazabilidad de los lotes de insectos, el seguimiento del stock, la gestión de ventas y la planificación de suministros, permitiendo una toma de decisiones más ágil confiable. En conclusión, este proyecto demuestra la importancia de una base de datos bien estructurada para optimizar los procesos, mejorar la eficiencia operativa y mantener la seguridad de la información dentro de una organización.

Referencias Bibliográficas

References

- [1] IBM. (2025, 16 junio). Bases de datos relacionales. IBM.
<https://www.ibm.com/mx-es/think/topics/relational-databases>
- [2] KYOCERA Document Solutions. (s. f.). El modelo relacional como utilidad para la calidad de los datos — Kyocera.
<https://www.kyoceradocumentsolutions.es/es/smarter-workspaces/insights-hub/articles/el-modelo-relacional-como-utilidad-para-la-calidad-de-los-datos.html#:~:text=Modelo>

- [3] Oracle. (2021, 18 junio). What is a Relational Database (RDBMS)? <https://www.oracle.com/es/database/what-is-a-relational-database/>: :text=El
- [4] ¿Qué es una base de datos relacional (RDBMS)? — Google Cloud. (s. f.). Google Cloud. <https://cloud.google.com/learn/what-is-a-relational-database?hl=es-419>