

# REPORTE DE PROYECTO FINAL

## NOMBRES DE LOS ALUMNOS:

Ian Abishai Ramírez Olvera

Vladimir Islas Batalla

Joal David Sanchez Vidal

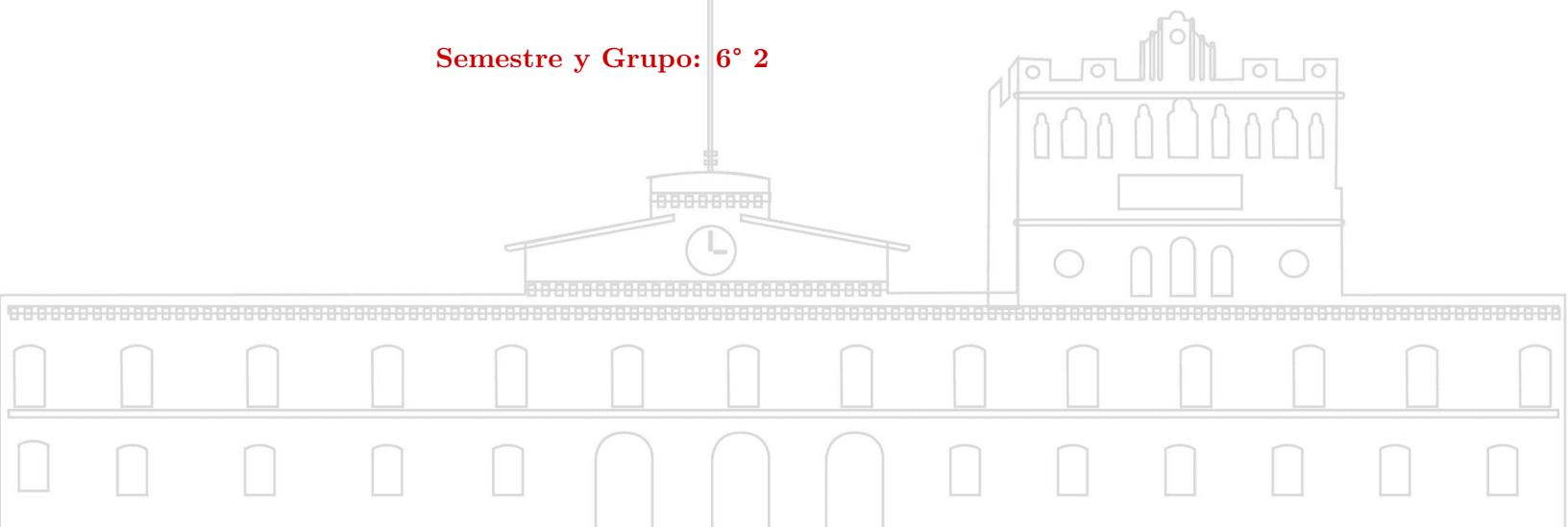
Yhojan Jesus Lozano Vazquez

Edwin Aaron Lopez Jimenez

Juan Pablo Ramos Hernandez

**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 planificar mejor la producción, evitando tanto los excedentes como la escasez de insetos 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, garantizado 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:

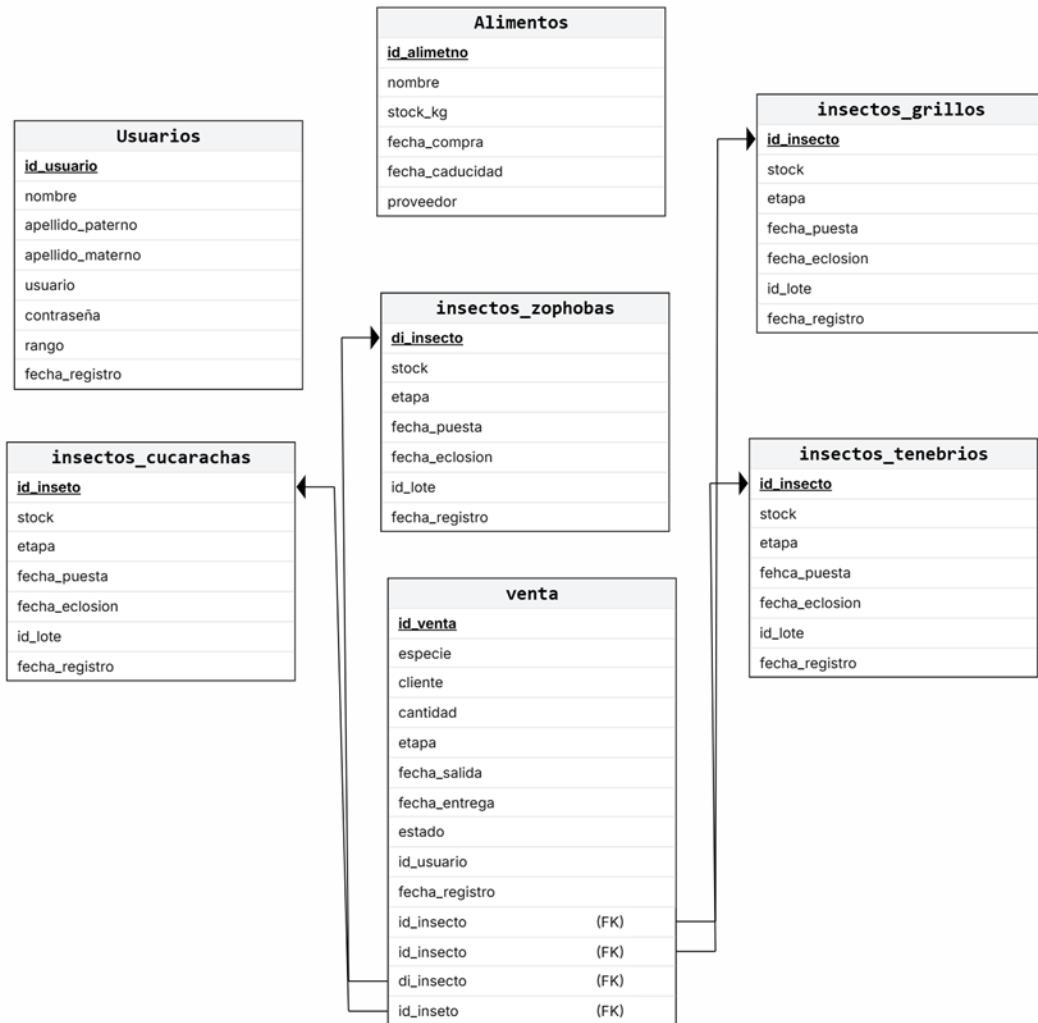
1. Administrador: Control total del sistema, gestión de usuarios, acceso a todos los módulos y reportes.
2. Gerente: Supervisión de inventarios y ventas, generación de reportes, acceso restringido a configuración de usuarios.
3. Operador: Manejo del inventario, registro de alimentos e insectos, actualización 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)
);

```

## Prueba de Tablas

– usuarios (3)

```

INSERT INTO usuarios (id_usuario, nombre,
    apellido_paterno, apellido_materno, usuario, contraseña, rango) VALUES
(1, 'Ana', 'López', 'García', 'ana.lopez', 'ana2025!', 'admin'),
(2, 'Carlos', 'Méndez', 'Ruiz', 'carlos.m', 'carlos2025!', 'vendedor'),
(3, 'María', 'Pérez', 'Hernández', 'maria.p', 'maria2025!', 'operador');

```

– insectos\_grillos (3)

```

INSERT INTO insectos_grillos (id_insecto, stock, etapa, fecha_puesta,
    fecha_eclosion, id_lote) VALUES
(1, 12000, 'adulto', '2025-08-01', '2025-08-20', 'GRI-2025-01'),
(2, 8000, 'larva', '2025-08-15', '2025-09-05', 'GRI-2025-02'),
(3, 5000, 'huevo', '2025-09-01', NULL, 'GRI-2025-03');

```

– insectos\_cucarachas (3)

```

INSERT INTO insectos_cucarachas (id_insecto, stock, etapa, fecha_puesta,
    fecha_eclosion, id_lote) VALUES
(1, 6000, 'adulto', '2025-07-20', '2025-08-10', 'CUC-2025-01'),
(2, 4000, 'larva', '2025-08-10', '2025-08-30', 'CUC-2025-02'),
(3, 2000, 'huevo', '2025-09-05', NULL, 'CUC-2025-03');

```

– insectos\_zophobas (3)

```
INSERT INTO insectos_zophobas (id_insecto, stock, etapa, fecha_puesta, fecha_eclosion, id_lote) VALUES  
(1, 9000, 'larva', '2025-07-25', NULL, 'ZOP-2025-01'),  
(2, 7000, 'adulto', '2025-07-15', '2025-08-05', 'ZOP-2025-02'),  
(3, 3000, 'huevo', '2025-08-25', NULL, 'ZOP-2025-03');
```

– insectos\_tenebrios (3)

```
INSERT INTO insectos_tenebrios (id_insecto, stock, etapa, fecha_puesta, fecha_eclosion, id_lote) VALUES  
(1, 11000, 'adulto', '2025-08-05', '2025-08-25', 'TEN-2025-01'),  
(2, 6000, 'larva', '2025-08-20', '2025-09-10', 'TEN-2025-02'),  
(3, 4000, 'huevo', '2025-09-10', NULL, 'TEN-2025-03');
```

– insectosventas (3) - id\_insecto = 1 de cada especie

```
INSERT INTO ventas (id_venta, id_insecto, especie, cantidad, etapa, fecha_salida, id_usuario) VALUES  
(1, 1, 'grillo', 2000, 'adulto', '2025-09-15', 1),  
(2, 1, 'cucaracha', 1000, 'adulto', '2025-09-16', 2),  
(3, 1, 'zophobas', 1500, 'larva', '2025-09-17', 1),  
(4, 1, 'tenebrio', 3000, 'adulto', '2025-09-18', 1),  
(5, 1, 'grillo', 2500, 'adulto', '2025-09-19', 2),  
(6, 1, 'cucaracha', 800, 'adulto', '2025-09-20', 1),  
(7, 1, 'zophobas', 2000, 'larva', '2025-09-21', 2),  
(8, 1, 'tenebrio', 4000, 'larva', '2025-09-22', 1),  
(9, 1, 'grillo', 1500, 'adulto', '2025-09-23', 2),  
(10, 1, 'cucaracha', 1200, 'adulto', '2025-09-24', 1),
```

## Insercion de datos(INSERT)

```
INSERT INTO usuarios (id_usuario, nombre, apellido_paterno, usuario, contraseña, rango, fecha_registro)
```

```
INSERT INTO usuarios (id_usuario, nombre, apellido_paterno, usuario, contraseña, 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;
```

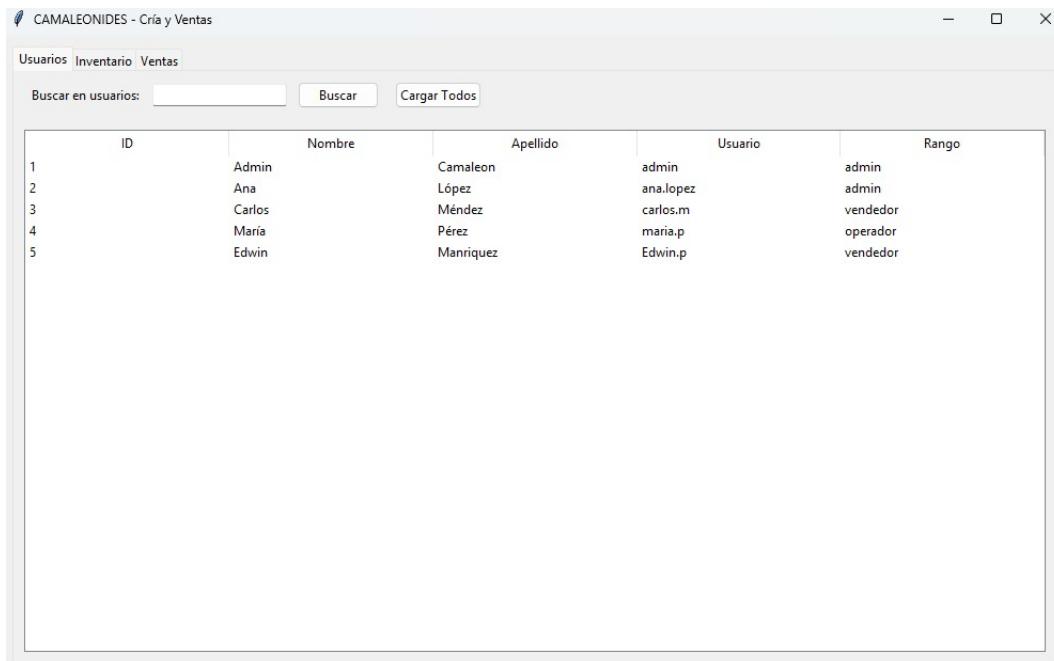
## Eliminación de datos (DELETE)

```
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);
```

## Salida en Python



The screenshot shows a Python application window titled "CAMALEONIDES - Cría y Ventas". The window has three tabs at the top: "Usuarios", "Inventario", and "Ventas". Below the tabs are three buttons: "Buscar" (Search), "Cargar Todos" (Load All), and a search input field labeled "Buscar en usuarios:". The main area is a table with the following data:

ID	Nombre	Apellido	Usuario	Rango
1	Admin	Camaleon	admin	admin
2	Ana	López	ana.lopez	admin
3	Carlos	Méndez	carlos.m	vendedor
4	Maria	Pérez	maria.p	operador
5	Edwin	Manriquez	Edwin.p	vendedor

CAMALEONIDES - Cria y Ventas

Usuarios Inventario Ventas

Buscar en insectos:  Buscar Cargar Todos

Especie	ID	Stock	Etapa
Grillos	1	12000	adulto
Grillos	2	8000	larva
Grillos	3	5000	huevo
Cucarachas	1	6000	adulto
Cucarachas	2	4000	larva
Cucarachas	3	2000	huevo
Zophobas	1	9000	larva
Zophobas	2	7000	adulto
Zophobas	3	3000	huevo
Tenebrios	1	11000	adulto
Tenebrios	2	6000	larva
Tenebrios	3	4000	huevo

CAMALEONIDES - Cria y Ventas

Usuarios Inventario Ventas

Buscar en ventas:  Buscar Cargar Todos

ID	Especie	Cantidad	Etapa	Fecha	Estado
10	cucaracha	1200	adulto	2025-09-24	pendiente
9	grillo	1500	adulto	2025-09-23	pendiente
8	tenebrio	4000	larva	2025-09-22	pendiente
7	zophobas	2000	larva	2025-09-21	pendiente
6	cucaracha	800	adulto	2025-09-20	pendiente
5	grillo	2500	adulto	2025-09-19	pendiente
4	tenebrio	3000	adulto	2025-09-18	pendiente
3	zophobas	1500	larva	2025-09-17	pendiente
2	cucaracha	1000	adulto	2025-09-16	pendiente
1	grillo	2000	adulto	2025-09-15	pendiente

## **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>