

# Mi Granja Inteligente

- Objetivo del Proyecto: Crear una base de datos para administrar los animales de una granja, donde se registren datos clave como especie, edad, sexo, peso, número de identificación, estado de salud.
- Animales en la Granja:
- Vacas 🐄
  - Ovejas 🐑
  - Corderos (crías de oveja) 🐑
  - Gallos 🐔
  - Gallinas 🐔
  - Toros 🐂
  - Conejos 🐰
  - Pavos 🦃
  - Cabras 🐐
  -

## 📁 Estructura del Repositorio

```
granja-digital/
├── README.md
├── datos/
│   └── datos_granja.sql
├── sql/
│   ├── crear_base_de_datos.sql
│   └── consultas_utiles.sql
├── imagenes/
│   └── diagrama_base_datos.png (opcional)
└── presentacion/
    └── presentacion_granja.pdf (o PowerPoint, si usas)
```

## 📝 Contenido del README.md

```
# 🐄 Proyecto Escolar: Granja Digital
```

```
Este proyecto simula la administración de una **granja digital** con diferentes tipos de animales, utilizando una base de datos creada con **MariaDB/MySQL**.
```

```
---
```

```
## 📈 Objetivos del Proyecto
```

- Simular la gestión de animales en una granja.
- Utilizar SQL para organizar y consultar datos.
- Practicar el uso de bases de datos relacionales.
- Presentar resultados de forma clara y ordenada.

```
---
```

```
## 🐄 Animales Incluidos
```

- Vacas 
- Toros 
- Ovejas 
- Corderos  
- Gallinas 
- Gallos 
- Conejos 
- Pavos 
- Cabras 

---

## ## 📈 Estructura de la Base de Datos

```
```sql
CREATE TABLE animales (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50),
    tipo_animal VARCHAR(30),
    sexo CHAR(1),
    edad INT,
    peso DECIMAL(6,2),
    estado_salud VARCHAR(30),
    fecha_ingreso DATE,
    observaciones TEXT
);
```

## 💾 Datos Simulados

```
mysql -u root -p granja < datos/datos_granja.sql
```

## 🔍 Consultas Útiles

```
-- Ver todos los animales
SELECT * FROM animales;

-- Contar animales por tipo
SELECT tipo_animal, COUNT(*) FROM animales GROUP BY tipo_animal;

-- Ver animales enfermos
SELECT * FROM animales WHERE estado_salud = 'Enferma';
```

## 📝 Cómo Ejecutar el Proyecto

Clona el repositorio:

```
git clone https://github.com/ademo/granja-digital.git  
cd granja-digital
```

Entra a MySQL y crea la base de datos:

```
SOURCE sql/crear_base_de_datos.sql;
```

Importa los datos:

```
SOURCE datos/datos_granja.sql;
```

## Autor

---

Tarik - Proyecto escolar realizado con fines educativos y de aprendizaje de bases de datos.

```
---  
  
##  Archivos a Subir  
  
### 1. `crear_base_de_datos.sql`  
```sql  
CREATE DATABASE granja;  
USE granja;  
  
CREATE TABLE animales (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(50),  
    tipo_animal VARCHAR(30),  
    sexo CHAR(1),  
    edad INT,  
    peso DECIMAL(6,2),  
    estado_salud VARCHAR(30),  
    fecha_ingreso DATE,  
    observaciones TEXT  
);
```

consultas\_utiles.sql

```
SELECT * FROM animales;
SELECT tipo_animal, COUNT(*) FROM animales GROUP BY tipo_animal;
SELECT * FROM animales WHERE estado_salud = 'Enferma';
SELECT tipo_animal, AVG(peso) FROM animales GROUP BY tipo_animal;
```

## Documentación de Vistas

- ◊ 1. vista\_animales\_enfermos

```
CREATE OR REPLACE VIEW vista_animales_enfermos AS
SELECT *
FROM animales
WHERE estado_salud = 'Enferma';
```

- ◊ 2. vista\_animales\_hembra

```
CREATE OR REPLACE VIEW vista_animales_hembra AS
SELECT *
FROM animales
WHERE sexo = 'H';
```

- ◊ 3. vista\_animales\_mayores5

```
CREATE OR REPLACE VIEW vista_animales_mayores5 AS
SELECT *
FROM animales
WHERE edad > 5;
```

- ◊ 4. vista\_animales\_saludables

```
CREATE OR REPLACE VIEW vista_animales_saludables AS
SELECT *
FROM animales
WHERE estado_salud = 'Saludable';
```

- ◊ 5. vista\_animales\_sin\_imagen

```
CREATE OR REPLACE VIEW vista_animales_sin_imagen AS
SELECT *
FROM animales
WHERE imagen IS NULL;
```

- ◊ 6. vista\_aves\_con\_problemas

```
CREATE OR REPLACE VIEW vista_aves_con_problemas AS
SELECT *
FROM animales
WHERE tipo_animal IN ('Gallina', 'Gallo') AND estado_salud <> 'Saludable';
```

- ◊ 7. vista\_bovinos\_antiguos

```
CREATE OR REPLACE VIEW vista_bovinos_antiguos AS
SELECT *
FROM animales
WHERE tipo_animal IN ('Toro', 'Vaca') AND fecha_ingreso < '2024-01-01';
```

- ◊ 8. vista\_cabras

```
CREATE OR REPLACE VIEW vista_cabras AS
SELECT *
FROM animales
WHERE tipo_animal = 'Cabra';
```

- ◊ 9. vista\_conejos\_jovenes\_saludables

```
CREATE OR REPLACE VIEW vista_conejos_jovenes_saludables AS
SELECT *
FROM animales
WHERE tipo_animal = 'Conejo' AND edad < 5 AND estado_salud = 'Saludable';
```

- ◊ 10. vista\_ingresados\_2025

```
CREATE OR REPLACE VIEW vista_ingresados_2025 AS
SELECT *
FROM animales
WHERE EXTRACT(YEAR FROM fecha_ingreso) = 2025;
```

## Evidencia Visual del Proyecto en Airtable

A continuación se presenta una serie de pantallazos que documentan visualmente la configuración actual del proyecto en Airtable. Estas capturas sirven como respaldo gráfico de las estructuras descritas previamente y

ayudan a comprender de forma rápida la organización y funcionamiento de la base de datos.

The screenshot shows the Airtable homepage with the following elements:

- Header:** Airtable logo, search bar ("Buscar..."), and keyboard shortcut "Ctrl K".
- Sidebar:**
  - Inicio:** Placeholder for workspace icons.
  - Todos los espacios de trabajo + >**
  - Plantillas y apps**
  - Marketplace**
- Main Content Area:**

## Inicio

**Consigue mayor potencia con el plan Team**  
Más registros. Más automatizaciones. Más personalización. Más Airtable.

**Mejorar**      **Comparar planes**

**Comenzar con IA**  
Convierte tu proceso en una app con datos e interfaces mediante IA.

**Empieza con plantillas**  
Selecciona una plantilla para empezar y personalízala sobre la marcha.

**Carga rápidamente**  
Migra fácilmente tus proyectos existentes en tan solo unos m

Abierto por ti ▾ Mostrar todos los tipos ▾

**Últimos 7 días**

  - Tienda Online de Animales de Granja**  
Base
  - Animales de Granja**  
Interfaz · Tienda Online de Animales de Granja
- Automation Editor Screenshot:** Shows the "Tienda Online de Animales de Granja" workspace. An automation named "Automation 1" is selected, which triggers "A una hora programada" (Every 1 day at 8:00am CEST). The configuration shows a trigger type of "A la hora programada" and a cronology of "Cada 1 día a las 8:00am" starting on "16/6/2025". A "Probar desencadenante" (Test trigger) button is visible.

The screenshot shows a Softr Studio application interface for a "Tienda Online de Animales de Granja". The main area displays a grid of animal records with columns for Actions, Nombre, Fecha de Ingreso, id, and Tipo de Animal. Each row contains an "Edit" button and a three-dot menu icon. The grid includes four entries: Rayo12 (2025-05-21, id 14, Pavo), Rayo71 (2023-07-06, id 35, Cabra), Rayo79 (2023-06-29, id 51, Cabra), and Bruno6 (2022-12-31, id 100, Gallo). A search bar at the top right says "Type here to search" and an "Add record" button is visible.

The screenshot shows an Airtable grid view for the same "Tienda Online de Animales de Granja" dataset. The grid has columns for Nombre, Estado de Salud, Fecha de Ingreso, Observaciones, Precio, Inventario, id, and Can. The data rows correspond to the entries in the Softr grid, with additional columns for Estado de Salud (e.g., Recuperando, Saludable) and Observaciones (e.g., Tiene tos, Muy activo). The Airtable interface includes a sidebar with various view options like Grid view, Filtros, and Grupos, and a bottom navigation bar with 100 registros.

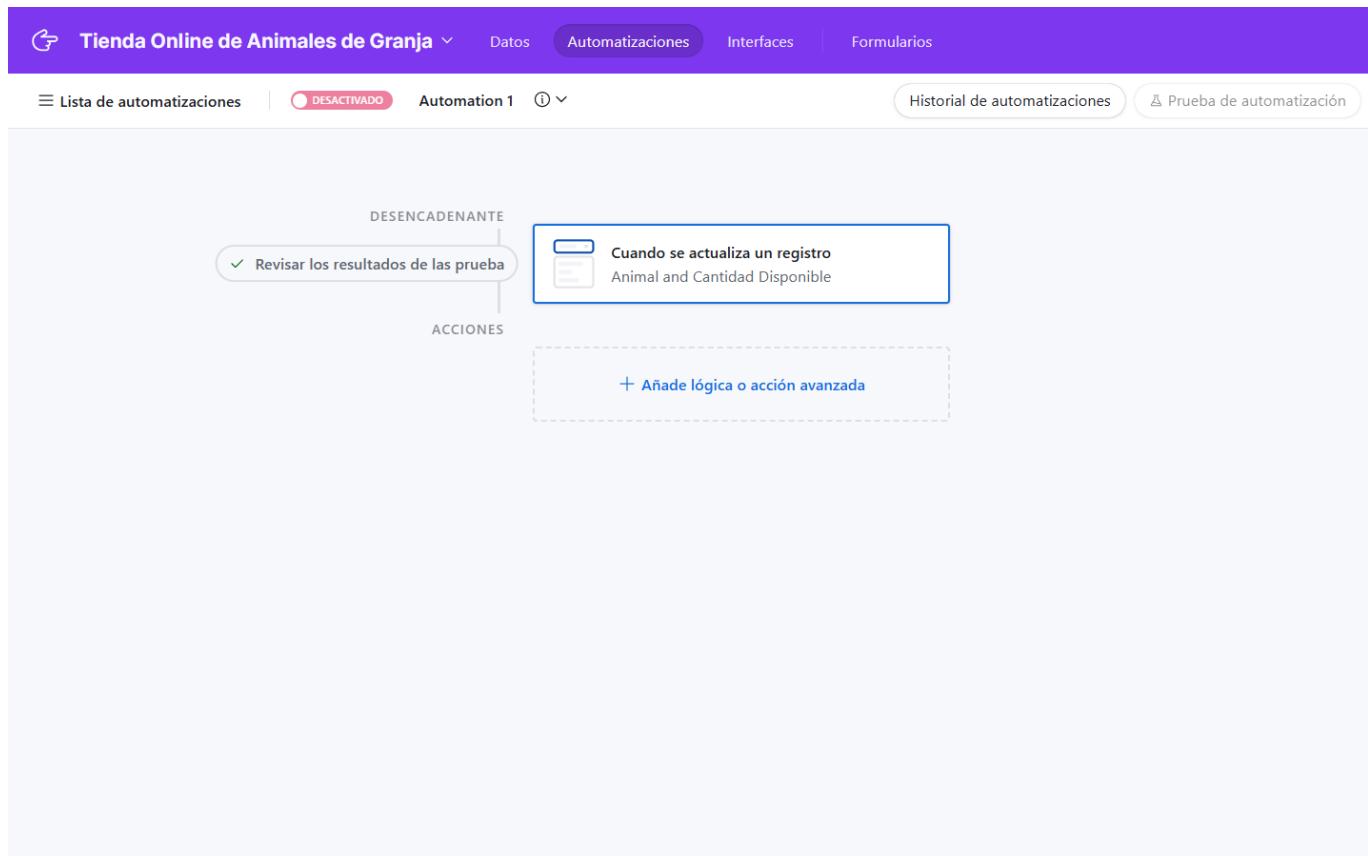
Tienda Online de Animales de Granja ▾ Dados Automatizaciones Interfaces Formularios ⚙ Ayuda Compartir Extensões Herramientas ▾

Animales de Granja ▾ Precios de Animales Información de Ventas Clientes Inventario de Productos Envíos Comentarios de Clientes Promociones +

Vistas Grid view Ocultar campos Filtro Grupo Clasificar Configurar color Compartir y sincronizar

	A Nombre	○ Tipo de Animal	○ Sexo	# Edad	# Peso	○ Estado de Salud	A Fecha de Ingreso	Observaciones	Precio	Inventario	A id
79	Rayo71	Oveja	H	1	573.25	Recuperando	2023-05-30	Tiene tos	845.70	13	79
80	Linda54	Cordero	M	2	143.18	Saludable	2024-05-09	Muy activo	146.10	10	80
81	Rayo61	Vaca	H	2	511.16	Saludable	2024-10-31	Animal reproductor	1322.00	9	81
82	Blanquita36	Oveja	H	2	765.99	Saludable	2025-03-24	-	254.80	12	82
83	Tornado42	Pavo	M	3	640.75	Recuperando	2023-04-07	Animal reproductor	158.25	8	83
84	Estrella16	Vaca	H	1	147.65	Saludable	2023-07-31	Tiene tos	414.90	6	84
85	Estrella69	Cordero	M	2	206.58	Saludable	2025-04-26	Animal reproductor	345.10	4	85
86	Tornado32	Conejo	M	1	448.27	Saludable	2023-01-30	Está preñada	599.99	7	86
87	Nieve13	Pavo	H	2	880.05	Saludable	2023-02-10	Muy ruidoso	498.00	15	87
88	Lola84	Vaca	H	3	233.56	Saludable	2025-05-31	-	269.99	10	88
89	Lola13	Toro	M	4	227.62	Recuperando	2025-04-23	-	144.75	5	89
90	Blanquita98	Toro	M	1	142.74	Saludable	2023-04-21	Muy ruidoso	376.25	14	90
91	Lola89	Gallina	H	3	255.24	Saludable	2023-08-21	Muy activo	1105.10	2	91
92	Bruno33	Vaca	H	2	742.92	Saludable	2023-03-11	Animal reproductor	297.50	11	92
93	Nieve74	Oveja	H	2	151.39	Recuperando	2023-07-12	Tiene tos	165.20	6	93
	+ Añadir...	Cabra	H	1	743.57	Recuperando	2023-11-26	Tiene tos	890.00	13	94

100 registros Suma 202 suma 48039.17



The screenshot shows a Airtable dashboard titled "Animales de Granja". The main view is "Overview" and the sub-view is "Animales de Granja dashboard". The table has the following columns: Nombre, Tipo de Animal, Sexo, Edad, Peso, and Estado de Salud. The data includes 15 entries such as Rayo12 (Pavo, M, 1, 469.05, Saludable), Rayo71 (Cabra, H, 1, 893.38, Saludable), Bruno6 (Gallo, M, 1, 471.19, Recuperando), etc.

Nombre	Tipo de Animal	Sexo	Edad	Peso	Estado de Salud
Rayo12	Pavo	M	1	469.05	Saludable
Rayo71	Cabra	H	1	893.38	Saludable
Rayo79	Cabra	H	1	313.46	Saludable
Bruno6	Gallo	M	1	471.19	Recuperando
Nieve53	Toro	M	4	684.22	Saludable
Estrella16	Vaca	H	1	147.65	Saludable
Lola99	Cabra	H	4	375.59	Saludable
Pancho6	Conejo	H	1	250.03	Saludable
Pancho30	Cordero	M	4	413.84	Saludable
Rayo8	Vaca	H	2	592.83	Saludable
Rayo12	Cabra	H	1	191.01	Saludable
Blanquita98	Toro	M	1	142.74	Saludable
Bruno27	Vaca	H	1	318.47	Saludable
Bruno13	Cabra	M	4	701.09	Recuperando
Nieve49	Cabra	M	3	901.41	Saludable

## 1. Relación entre tablas

### ## 📈 Relación entre tablas (Modelo Entidad-Relación)

- `animales` → tabla principal con la información de cada animal
- `inventario` → almacena el stock actual por tipo de animal
- `ventas` → registro de ventas realizadas
- `precios` → precios históricos de cada tipo de animal
- `envíos` → control de entregas al cliente
- `comentarios\_clientes` → opiniones recibidas por cada venta
- `promociones` → descuentos activos según especie o temporada

## 2. Scripts incluidos

### ## ⚡ Scripts incluidos

- `datos\_granja.sql`: Inserta los datos de los 100 animales
- `dump\_granja.sql`: Exportación completa de la base de datos
- `views.sql`: Contiene las vistas creadas en PostgreSQL para consultas avanzadas
- `airtable\_export.csv`: Exportación para cargar datos en Airtable

## 3. Automatizaciones programadas

### ## 🛡️ Automatizaciones

Se creó una automatización en Airtable que:

- Se ejecuta cada día a las 08:00
- Revisa la cantidad disponible
- Activa promociones si hay exceso de stock
- Envía notificación en caso de inventario bajo

## 4. Ejemplo de consulta SQL usada

```
## 🔎 Ejemplo de consulta SQL

```sql
-- Obtener animales con menos de 2 unidades disponibles
SELECT a.nombre, i.cantidad_disponible
FROM animales a
JOIN inventario i ON a.id = i.animal_id
WHERE i.cantidad_disponible < 2;
```

## Conclusión

Este proyecto fue una excelente oportunidad para aplicar conocimientos de bases de datos, automatización y visualización de información. A través de la simulación de una granja, logré construir un sistema completo que incluye la gestión de animales, inventario, precios, ventas y más, utilizando herramientas reales del mundo profesional como MariaDB, PostgreSQL, DBeaver y Airtable.

Este trabajo no solo refleja habilidades técnicas, sino también organización, planificación y visión de proyecto. Estoy orgulloso del resultado y abierto a seguir mejorándolo en el futuro.

Gracias por visitar este repositorio  