



JAGUAIA PARK

MySQL

Implementación

BASES DE DATOS
UD3_PRACTICA 3

ANDREA GUTIERREZ
ALEJANDRA VELÁSICO
FRANCELLA ROJAS



ÍNDICE

ÍNDICE	2
INTRODUCCIÓN.....	3
DESARROLLO	4
Mejoras solicitadas para implementación en SQL	4
1. Creación de una nueva entidad: TRATAMIENTO	4
2. Inclusión de dos relaciones N a N asociadas al nuevo TRATAMIENTO	4
3. Corrección de cardinalidad: relación N a N entre ANIMAL y VETERINARIO	5
Necesidad de una Base de Datos	6
Modelo Relacional Mejoras:.....	7
Implementación MySQL	8
Código MySQL	9
Bibliografía.....	11



INTRODUCCIÓN

La empresa que hemos creado lleva el nombre de “JAGUAIA”, una fusión entre Jaguar y Gaia (la diosa de la Tierra), en representación del equilibrio entre la fuerza animal y la naturaleza. Elegimos este nombre porque el jaguar es uno de nuestros animales favoritos y simboliza poder, respeto y conexión con el entorno natural.

JAGUAIA PARK es un parque de conservación y refugio de fauna silvestre ubicado en el norte de España, dedicado a la preservación, rehabilitación y estudio de especies animales provenientes de distintos ecosistemas. El parque desarrolla su labor a través de programas de conservación, donaciones privadas y colaboraciones científicas, contando con un equipo profesional compuesto por veterinarios, encargados y personal administrativo.

En el presente proyecto, realizaremos una serie de mejoras en el sistema de gestión de datos del parque, con el objetivo de optimizar la organización de la información relacionada con los animales, su hábitat y las actividades del personal. Estas mejoras permitirán una administración más eficiente, un control más preciso de los cuidados y un soporte estructurado para la investigación y la educación ambiental que promueve el zoológico.



DESARROLLO

Mejoras solicitadas para implementación en SQL

A partir del feedback recibido en la última presentación y tras revisar el modelo entregado en la UD1, se identificaron varias áreas de mejora necesarias para lograr una base de datos más completa, escalable y ajustada a la realidad operativa de un parque de conservación. Para esta nueva entrega se han implementado las siguientes mejoras:

1. Creación de una nueva entidad: TRATAMIENTO

En el modelo original, la información relacionada con medicación y procedimientos veterinarios estaba limitada a la entidad Consulta_Veterinaria. Sin embargo, en un entorno real un tratamiento puede ser aplicado en distintas fechas, por distintos veterinarios y a varios animales.

Por ello se incorpora la nueva entidad TRATAMIENTO, que permite registrar de forma estructurada:

Nombre del tratamiento

Descripción

Duración estimada

Frecuencia

Observaciones

Esta entidad añade profundidad al modelo médico y mejora la trazabilidad clínica.

2. Inclusión de dos relaciones N a N asociadas al nuevo TRATAMIENTO

Para reflejar adecuadamente la realidad del trabajo veterinario, el tratamiento requiere dos relaciones de muchos a muchos:

a) ANIMAL ↔ TRATAMIENTO

Un animal puede recibir múltiples tratamientos a lo largo de su vida, y un mismo tratamiento puede aplicarse a varios animales.

Se crea la tabla intermedia ANIMAL_TRATAMIENTO, que almacena además fechas, estado del tratamiento y observaciones.

b) VETERINARIO ↔ TRATAMIENTO



Diversos veterinarios pueden administrar un mismo tratamiento, y un veterinario administra numerosos tratamientos.

Se incorpora la tabla intermedia VET_TRATAMIENTO, que permite registrar quién aplica cada tratamiento y cuándo.

3. Corrección de cardinalidad: relación N a N entre ANIMAL y VETERINARIO

En el modelo original, cada animal quedaba asociado a un único veterinario, lo cual no refleja el funcionamiento real del parque.

Se modifica esta relación para convertirla en una relación muchos a muchos, implementada mediante la nueva tabla intermedia ATENDER, que registra:

- ID del veterinario
- ID del animal
- Fecha
- Diagnóstico
- Tratamiento aplicado
- Observaciones

Con esta mejora, cualquier animal puede ser atendido por más de un veterinario y cualquier veterinario puede tratar a múltiples animales.

Impacto global de las mejoras

Estas modificaciones fortalecen el modelo lógico y lo alinean mejor con escenarios reales de conservación, permitiendo:

- Un registro clínico completo
- Un historial médico rastreable
- Mayor integridad y coherencia entre entidades
- Un diseño preparado para su implementación en SQL con integridad referencial sólida.



Necesidad de una Base de Datos

Debido al crecimiento del parque y a la incorporación de nuevas especies, el sistema manual de registro —basado en fichas en papel y hojas de cálculo aisladas— se volvió insuficiente, poco fiable y propenso a errores. La falta de integración entre las áreas veterinaria, administrativa y operativa dificultaba el seguimiento adecuado de cada animal y hacía imposible mantener un control preciso sobre sus cuidados y tratamientos.

Por ello, el parque requiere una base de datos centralizada y escalable que permita:

- Gestionar de forma estructurada al personal, incluyendo veterinarios, encargados y sus funciones específicas dentro del parque.
- Registrar de manera completa el control médico de los animales, incorporando consultas veterinarias, diagnósticos y un historial detallado de tratamientos, permitiendo que un animal pueda recibir atención por parte de múltiples veterinarios.
- Planificar y almacenar información sobre las rutinas de cuidado diario (alimentación, peso, revisiones periódicas), así como el personal encargado de su ejecución.
- Asociar cada animal con su especie y con el hábitat natural que ocupa dentro del parque, garantizando un seguimiento ambiental adecuado.
- Administrar las donaciones de patrocinadores que apoyan económicamente a determinados animales o programas de conservación.

La implementación de esta nueva base de datos proporciona información actualizada y precisa para apoyar las decisiones clínicas, logísticas y económicas del parque, favoreciendo un manejo profesional y sostenible de todas las áreas involucradas.



Modelo Relacional Mejoras:

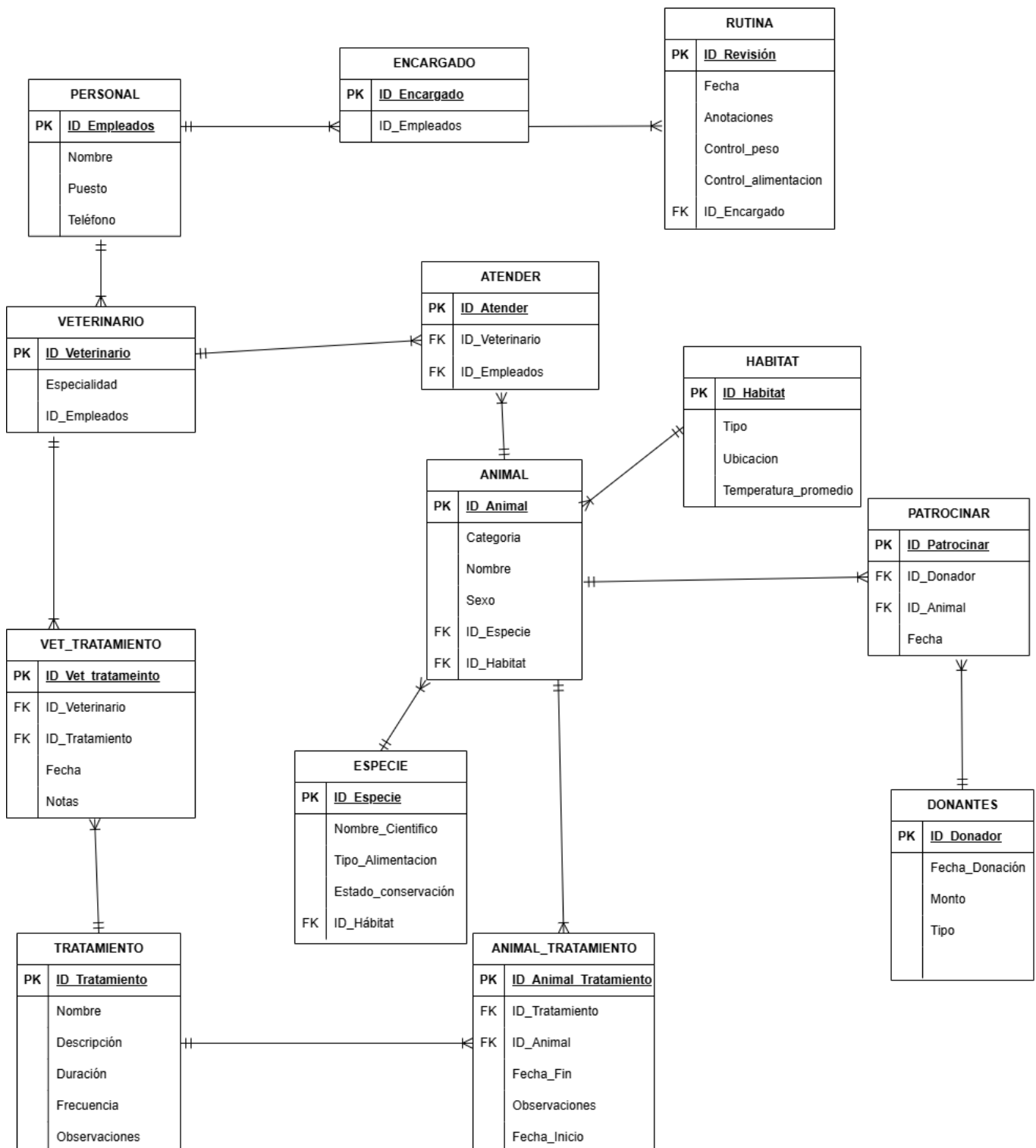


Ilustración 1Draw.io Modelo Relacional



Implementación MySQL

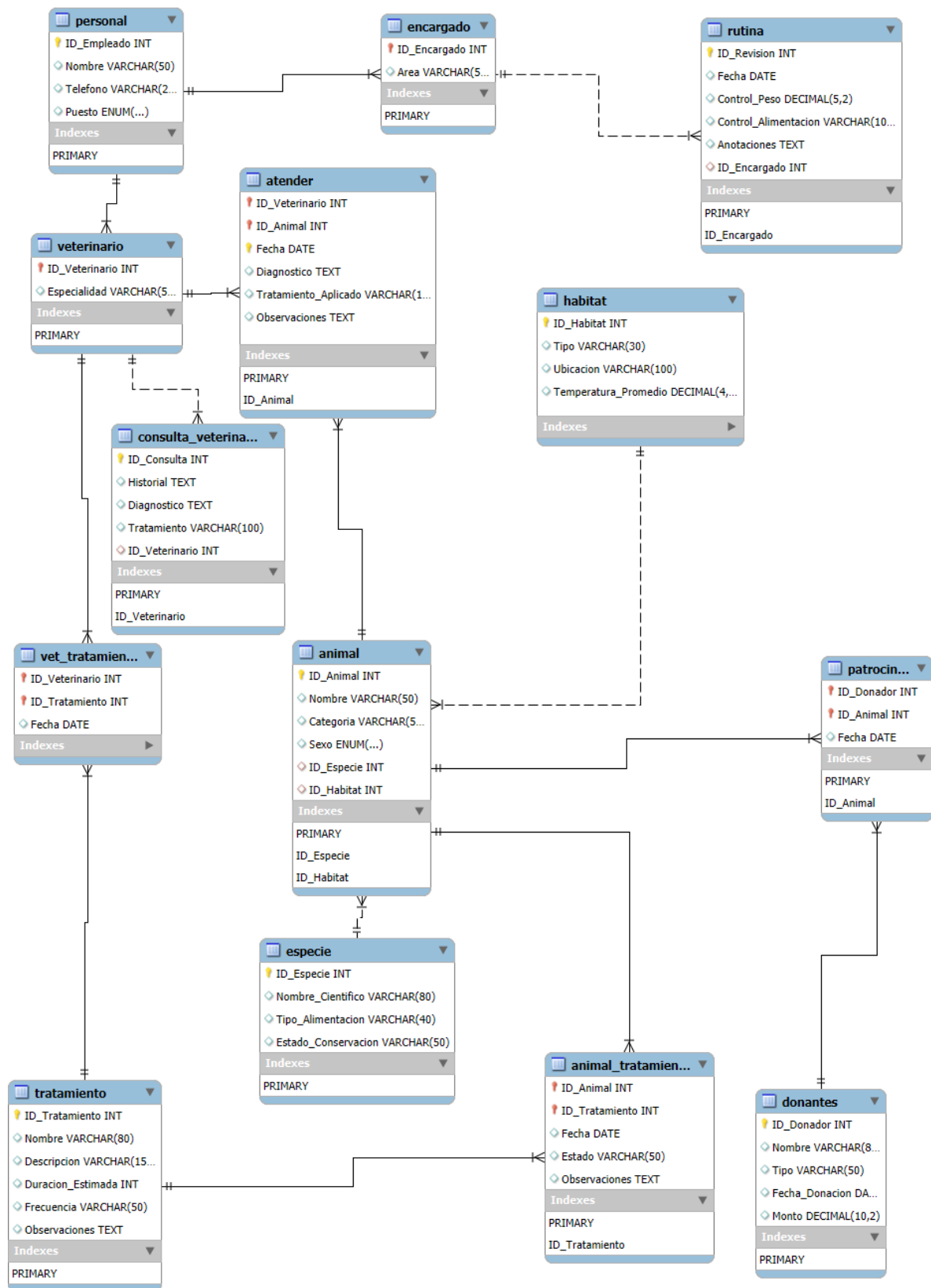


Ilustración 2MySQL WorkBench_JaguaiaPark



Código MySQL

```
CREATE DATABASE IF NOT EXISTS jaguaia_park_fin;  
USE jaguaia_park_fin;
```

```
CREATE TABLE Habitat (  
  ID_Habitat INT AUTO_INCREMENT PRIMARY KEY,  
  Tipo VARCHAR(30),  
  Ubicacion VARCHAR(100),  
  Temperatura_Promedio DECIMAL(4,1)  
);
```

```
CREATE TABLE Especie (  
  ID_Especie INT AUTO_INCREMENT PRIMARY KEY,  
  Nombre_Cientifico VARCHAR(80),  
  Tipo_Alimentacion VARCHAR(40),  
  Estado_Conservacion VARCHAR(50)  
);
```

```
CREATE TABLE Animal (  
  ID_Animal INT AUTO_INCREMENT PRIMARY KEY,  
  Nombre VARCHAR(50),  
  Categoría VARCHAR(50),  
  Sexo ENUM('Macho', 'Hembra'),  
  ID_Especie INT,  
  ID_Habitat INT,  
  FOREIGN KEY (ID_Especie) REFERENCES Especie(ID_Especie),  
  FOREIGN KEY (ID_Habitat) REFERENCES Habitat(ID_Habitat)  
);
```

```
CREATE TABLE Donantes (  
  ID_Donador INT AUTO_INCREMENT PRIMARY KEY,  
  Nombre VARCHAR(80),  
  Tipo VARCHAR(50),  
  Fecha_Donacion DATE,  
  Monto DECIMAL(10,2)  
);
```

```
CREATE TABLE Personal (  
  ID_Empleado INT AUTO_INCREMENT PRIMARY KEY,  
  Nombre VARCHAR(50),  
  Teléfono VARCHAR(20),  
  Puesto ENUM('Veterinario', 'Encargado')  
);
```

```
CREATE TABLE Veterinario (  
  ID_Veterinario INT PRIMARY KEY,  
  Especialidad VARCHAR(50),  
  FOREIGN KEY (ID_Veterinario) REFERENCES Personal(ID_Empleado)  
);
```

```
CREATE TABLE Encargado (  
  ID_Encargado INT PRIMARY KEY,  
  Area VARCHAR(50),  
  FOREIGN KEY (ID_Encargado) REFERENCES Personal(ID_Empleado)  
);
```

```
CREATE TABLE Consulta_Veterinaria (  
  ID_Consulta INT AUTO_INCREMENT PRIMARY KEY,  
  Historial TEXT,  
  Diagnostico TEXT,
```



```

Tratamiento VARCHAR(100),
ID_Veterinario INT,
FOREIGN KEY (ID_Veterinario) REFERENCES Veterinario(ID_Veterinario)
);

CREATE TABLE Rutina (
  ID_Revision INT AUTO_INCREMENT PRIMARY KEY,
  Fecha DATE,
  Control_Peso DECIMAL(5,2),
  Control_Alimentacion VARCHAR(100),
  Anotaciones TEXT,
  ID_Encargado INT,
  FOREIGN KEY (ID_Encargado) REFERENCES Encargado(ID_Encargado)
);

CREATE TABLE Tratamiento (
  ID_Tratamiento INT AUTO_INCREMENT PRIMARY KEY,
  Nombre VARCHAR(80),
  Descripcion VARCHAR(150),
  Duracion_Estimada INT,
  Frecuencia VARCHAR(50),
  Observaciones TEXT
);

CREATE TABLE Animal_Tratamiento (
  ID_Animal INT,
  ID_Tratamiento INT,
  Fecha DATE,
  Estado VARCHAR(50),
  Observaciones TEXT,
  PRIMARY KEY (ID_Animal, ID_Tratamiento),
  FOREIGN KEY (ID_Animal) REFERENCES Animal(ID_Animal),
  FOREIGN KEY (ID_Tratamiento) REFERENCES Tratamiento(ID_Tratamiento)
);

CREATE TABLE Vet_Tratamiento (
  ID_Veterinario INT,
  ID_Tratamiento INT,
  Fecha DATE,
  PRIMARY KEY (ID_Veterinario, ID_Tratamiento),
  FOREIGN KEY (ID_Veterinario) REFERENCES Veterinario(ID_Veterinario),
  FOREIGN KEY (ID_Tratamiento) REFERENCES Tratamiento(ID_Tratamiento)
);

CREATE TABLE Atender (
  ID_Veterinario INT,
  ID_Animal INT,
  Fecha DATE,
  Diagnostico TEXT,
  Tratamiento_Aplicado VARCHAR(100),
  Observaciones TEXT,
  PRIMARY KEY (ID_Veterinario, ID_Animal, Fecha),
  FOREIGN KEY (ID_Veterinario) REFERENCES Veterinario(ID_Veterinario),
  FOREIGN KEY (ID_Animal) REFERENCES Animal(ID_Animal)
);

CREATE TABLE Patrocinar (
  ID_Donador INT,
  ID_Animal INT,
  Fecha DATE,
  PRIMARY KEY (ID_Donador, ID_Animal),
  FOREIGN KEY (ID_Donador) REFERENCES Donantes(ID_Donador),
  FOREIGN KEY (ID_Animal) REFERENCES Animal(ID_Animal)
);

```



Bibliografía

No se ha utilizado Bibliografía en este trabajo.

