

PROYECTO FINAL - PARQUE NATURAL



PRIMERA PRE- ENTREGA

CURSO SQL

FECHA: 02/07/2024

INTEGRANTE (Nombre y apellido, Mail):

Agustina Luz Pellizzaro agusluzpelli@gmail.com

ÍNDICE

1. INTRODUCCIÓN	2
2. OBJETIVO	2
3. HERRAMIENTAS UTILIZADAS	2
4. TEMÁTICA DE LA BASE DE DATOS	2
5. DER	3
6. LISTADO DE TABLAS	4
7. CREACIÓN DE TABLAS	9

1. INTRODUCCIÓN

En el presente proyecto se expone y explica la creación de una base de datos relacional, con sus respectivos componentes, utilizando MySQL como sistema de gestión de la misma. La base de datos permite representar y dar soporte a los diferentes parques nacionales y reservas naturales que posean una diversidad de flora y fauna que están abiertos al público y se promueva el turismo y una variedad de actividades recreativas y educativas. Se explicará a lo largo del documento cuál fue el proceso completo para crear la base de datos, carga de datos, creación de diferentes procedimientos y diferentes objetos para poder administrar la reserva natural.

2. OBJETIVO

El propósito de este proyecto es poder desarrollar de manera completa una base de datos basada en un dominio real como es un parque nacional, priorizando la utilidad que el modelo le puede aportar a los diferentes actores involucrados. Esto permitirá poder aplicar los conocimientos adquiridos durante el curso "SQL" de Coderhouse en un caso práctico.

3. HERRAMIENTAS UTILIZADAS

Para el desarrollo del siguiente proyecto, se utilizaron las herramientas:

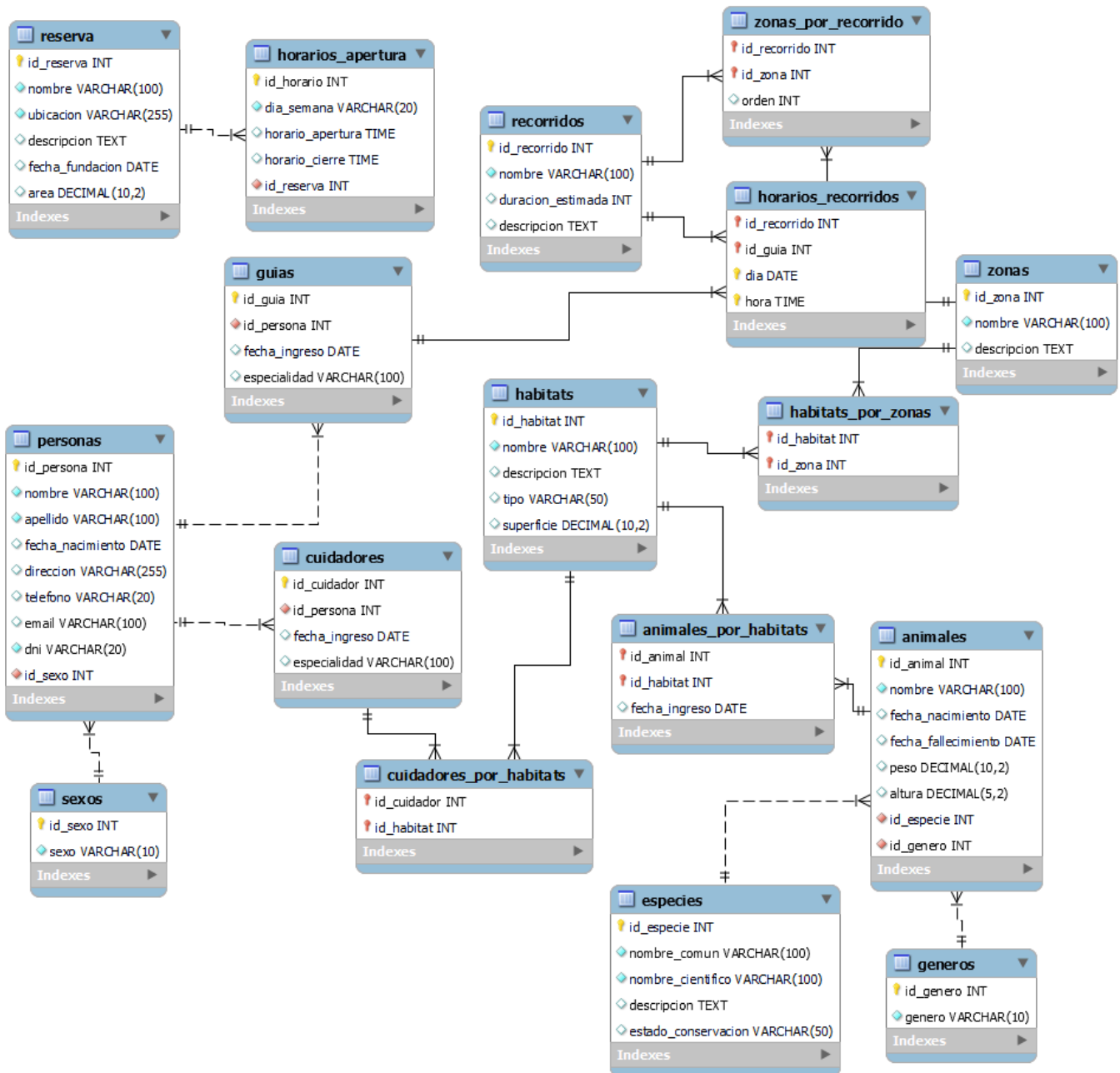
- MySQL Workbench
- Github: https://github.com/aguspellizzaro/proyecto_final_reserva_natural_pellizzaro

4. TEMÁTICA DE LA BASE DE DATOS

La temática es una base de datos que permita representar cualquier parque natural o reserva natural que posea nuestro país, donde el objetivo de las mismas sea poder brindar información a la población sobre las diferentes características del parque y a su vez colaborar a la gestión y conservación del parque. Para este tipo de entidades se suelen manejar grandes volúmenes de datos relacionados con la biodiversidad, especialmente con la fauna del lugar y las diferentes actividades que se pueden realizar. Por lo tanto, decidí crear una base de datos que contenga información sobre las diferentes especies de animales, sus hábitats, los recorridos disponibles en el parque, los guías y cuidadores, los horarios de apertura, y los visitantes. Una base de datos que incluya toda esta información y más puede ser muy útil para recolectar y acceder fácilmente a datos clave, permitiendo así descubrir insights valiosos para la gestión y conservación del parque, y al mismo tiempo proveer información a la población y visitas del parque sobre la importancia del cuidado de la biodiversidad e información educativa.

5. DER

Se desarrolló el diagrama de entidad relación en el MySQL Workbench. Se adjunta a continuación la imagen:



El archivo .mwb y el archivo .png se encuentran dentro del github.

6. LISTADO DE TABLAS

El listado de tablas se encuentra dentro del github con formato .excel.

Se adjunta de igual forma tabla:

Tabla	Descripción	Columna	Descripción	Tipo de Dato	Primary Key	Foreign Key	Uniq ue	Not Null	Auto Increment
reserva	Contiene información sobre las reservas naturales	id_reserva	Identificador único de la reserva	INT	Sí			Sí	Sí
		nombre	Nombre de la reserva	VARCHAR(100)				Sí	
		ubicacion	Ubicación de la reserva	VARCHAR(255)				Sí	
		descripcion	Descripción de la reserva	TEXT					
		fecha_fundacion	Fecha de fundación de la reserva	DATE					
		area	Área de la reserva en metros cuadrados	DECIMAL(10, 2)					
generos	Contiene los géneros de animales	id_genero	Identificador único del género	INT	Sí			Sí	Sí
		genero	Nombre del género	VARCHAR(10)				Sí	
sexos	Contiene los sexos de personas y animales	idsexo	Identificador único del sexo	INT	Sí			Sí	Sí
		sexo	Nombre del sexo	VARCHAR(10)				Sí	
especies	Contiene las especies de animales	id_especie	Identificador único de la especie	INT	Sí			Sí	Sí
		nombre_comun	Nombre común de la especie	VARCHAR(100)				Sí	
		nombre_cientifico	Nombre científico de la especie	VARCHAR(100)				Sí	

PROYECTO FINAL - CURSO SQL - CODER HOUSE - AGUSTINA PELLIZZARO

		descripcion	Descripción de la especie	TEXT					
		estado_conservacion	Estado de conservación de la especie	VARCHAR(50)					
animales	Contiene información sobre los animales	id_animal	Identificador único del animal	INT	Sí			Sí	Sí
		nombre	Nombre del animal	VARCHAR(100)				Sí	
		fecha_nacimiento	Fecha de nacimiento del animal	DATE					
		fecha_fallecimiento	Fecha de fallecimiento del animal	DATE					
		peso	Peso del animal en kg	DECIMAL(10, 2)					
		altura	Altura del animal en metros	DECIMAL(5, 2)					
		id_especie	Identificador de la especie	INT		Sí (especies.id_especie)		Sí	
		id_genero	Identificador del género del animal	INT		Sí (generos.id_genero)		Sí	
habitats	Contiene información sobre los hábitats de las reservas	id_habitat	Identificador único del hábitat	INT	Sí			Sí	Sí
		nombre	Nombre del hábitat	VARCHAR(100)				Sí	
		descripcion	Descripción del hábitat	TEXT					
		tipo	Tipo de hábitat	VARCHAR(50)					
		superficie	Superficie del hábitat en metros cuadrados	DECIMAL(10, 2)					
zonas	Contiene información sobre	id_zona	Identificador único de la zona	INT	Sí			Sí	Sí

PROYECTO FINAL - CURSO SQL - CODER HOUSE - AGUSTINA PELLIZZARO

	las zonas dentro de las reservas								
		nombre	Nombre de la zona	VARCHAR(100)				Sí	
		descripcion	Descripción de la zona	TEXT					
habitats_por_zonas	Relaciona hábitats con zonas	id_habitat	Identificador del hábitat	INT	Sí	Sí (habitats.id_habitat)			
		id_zona	Identificador de la zona	INT	Sí	Sí (zonas.id_zona)			
animales_por_habitats	Relaciona animales con hábitats	id_animal	Identificador del animal	INT	Sí	Sí (animales.id_animal)			
		id_habitat	Identificador del hábitat	INT	Sí	Sí (habitats.id_habitat)			
		fecha_ingreso	Fecha de ingreso del animal al hábitat	DATE					
personas	Contiene información sobre las personas	id_persona	Identificador único de la persona	INT	Sí			Sí	Sí
		nombre	Nombre de la persona	VARCHAR(100)				Sí	
		apellido	Apellido de la persona	VARCHAR(100)				Sí	
		fecha_nacimiento	Fecha de nacimiento de la persona	DATE					
		direccion	Dirección de la persona	VARCHAR(255)					
		telefono	Teléfono de la persona	VARCHAR(20)					
		email	Correo electrónico de la persona	VARCHAR(100)					

PROYECTO FINAL - CURSO SQL - CODER HOUSE - AGUSTINA PELLIZZARO

		dni	Documento nacional de identidad	VARCHAR(20)			Sí	Sí	
		id_sexo	Identificador del sexo de la persona	INT		Sí (sexos.id_sexo)		Sí	
cuidadores	Contiene información sobre los cuidadores de las reservas	id_cuidador	Identificador único del cuidador	INT	Sí			Sí	Sí
		id_persona	Identificador de la persona	INT		Sí (personas.id_persona)			
		fecha_ingreso	Fecha de ingreso como cuidador	DATE					
		especialidad	Especialidad del cuidador	VARCHAR(100)					
cuidadores_por_habitats	Relaciona cuidadores con hábitats	id_cuidador	Identificador del cuidador	INT	Sí	Sí (cuidadores.id_cuidador)			
		id_habitat	Identificador del hábitat	INT	Sí	Sí (habitats.id_habitat)			
guias	Contiene información sobre los guías de los recorridos	id_guia	Identificador único del guía	INT	Sí			Sí	Sí
		id_persona	Identificador de la persona	INT		Sí (personas.id_persona)			
		fecha_ingreso	Fecha de ingreso como guía	DATE					
		especialidad	Especialidad del guía	VARCHAR(100)					
recorridos	Contiene información sobre los recorridos dentro de las reservas	id_recorrido	Identificador único del recorrido	INT	Sí			Sí	Sí

PROYECTO FINAL - CURSO SQL - CODER HOUSE - AGUSTINA PELLIZZARO

		nombre	Nombre del recorrido	VARCHAR(100)				Sí	
		duracion_estimada	Duración estimada del recorrido en minutos	INT					
		descripcion	Descripción del recorrido	TEXT					
zonas_por_recorrido	Relaciona zonas con recorridos	id_recorrido	Identificador del recorrido	INT	Sí	Sí (recorridos.id_recorrido)			
		id_zona	Identificador de la zona	INT	Sí	Sí (zonas.id_zona)			
		orden	Orden del recorrido dentro de la zona	INT					
horarios_apertura	Contiene los horarios de apertura de las reservas	id_horario	Identificador único del horario	INT	Sí			Sí	Sí
		dia_semana	Día de la semana	VARCHAR(20)				Sí	
		horario_apertura	Hora de apertura	TIME					
		horario_cierre	Hora de cierre	TIME					
		id_reserva	Identificador de la reserva	INT		Sí (reserva.id_reserva)		Sí	
horarios_recorridos	Contiene los horarios de recorridos con guías	id_recorrido	Identificador del recorrido	INT		Sí (recorridos.id_recorrido)			
		id_guia	Identificador del guía	INT		Sí (guias.id_guia)		Sí	
		dia	Día del horario	DATE					
		hora	Hora del horario	TIME					

7. CREACIÓN DE TABLAS

El archivo .sql con el script de creación de tablas se encuentra dentro del github:

Se adjunta de igual script:

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

```
CREATE SCHEMA IF NOT EXISTS reservas_naturales;
USE reservas_naturales;
```

-- Tabla de Reserva

```
CREATE TABLE IF NOT EXISTS reserva (
  id_reserva INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  ubicacion VARCHAR(255) NOT NULL,
  descripcion TEXT,
  fecha_fundacion DATE,
  area DECIMAL(10, 2) -- Área en metros cuadrados
);
```

-- Tabla de Generos

```
CREATE TABLE IF NOT EXISTS generos (
  id_genero INT AUTO_INCREMENT PRIMARY KEY,
  genero VARCHAR(10) NOT NULL
);
```

-- Tabla de Sexos

```
CREATE TABLE IF NOT EXISTS sexos (
  id_sexo INT AUTO_INCREMENT PRIMARY KEY,
  sexo VARCHAR(10) NOT NULL
);
```

-- Tabla de Especies de Animales

```
CREATE TABLE IF NOT EXISTS especies (
  id_especie INT AUTO_INCREMENT PRIMARY KEY,
  nombre_comun VARCHAR(100) NOT NULL,
  nombre_cientifico VARCHAR(100) NOT NULL,
  descripcion TEXT,
  estado_conservacion VARCHAR(50)
);
```

```
CREATE INDEX idx_especies_nombre_comun ON especies(nombre_comun);
CREATE INDEX idx_especies_nombre_cientifico ON especies(nombre_cientifico);
```

-- Tabla de Animales

```
CREATE TABLE IF NOT EXISTS animales (
```

```

id_animal INT AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(100) NOT NULL,
fecha_nacimiento DATE,
fecha_fallecimiento DATE,
peso DECIMAL(10, 2), -- Peso del animal en kg
altura DECIMAL(5, 2), -- Altura del animal en metros
id_especie INT NOT NULL,
id_genero INT NOT NULL,
CONSTRAINT fk_animales_especie FOREIGN KEY (id_especie) REFERENCES especies(id_especie),
CONSTRAINT fk_animales_genero FOREIGN KEY (id_genero) REFERENCES generos(id_genero)
);

CREATE INDEX idx_animales_nombre ON animales(nombre);

-- Tabla de Habitats
CREATE TABLE IF NOT EXISTS habitats (
    id_habitat INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT,
    tipo VARCHAR(50),
    superficie DECIMAL(10, 2) -- Superficie en metros cuadrados
);

CREATE INDEX idx_habitats_nombre ON habitats(nombre);

-- Tabla de Zonas
CREATE TABLE IF NOT EXISTS zonas (
    id_zona INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT
);

CREATE INDEX idx_zonas_nombre ON zonas(nombre);

-- Tabla de Habitats por Zonas
CREATE TABLE IF NOT EXISTS habitats_por_zonas (
    id_habitat INT,
    id_zona INT,
    PRIMARY KEY (id_habitat, id_zona),
    CONSTRAINT fk_habitats_por_zonas_habitat FOREIGN KEY (id_habitat) REFERENCES
habitats(id_habitat),
    CONSTRAINT fk_habitats_por_zonas_zona FOREIGN KEY (id_zona) REFERENCES zonas(id_zona)
);

-- Tabla de Animales por Habitats
CREATE TABLE IF NOT EXISTS animales_por_habitats (
    id_animal INT,

```

```

id_habitat INT,
fecha_ingreso DATE,
PRIMARY KEY (id_animal, id_habitat),
        CONSTRAINT fk_animales_por_habitats_animal FOREIGN KEY (id_animal) REFERENCES
animales(id_animal),
        CONSTRAINT fk_animales_por_habitats_habitat FOREIGN KEY (id_habitat) REFERENCES
habitats(id_habitat)
);

```

-- Tabla de Personas

```

CREATE TABLE IF NOT EXISTS personas (
    id_persona INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    fecha_nacimiento DATE,
    direccion VARCHAR(255),
    telefono VARCHAR(20),
    email VARCHAR(100),
    dni VARCHAR(20) UNIQUE NOT NULL,
    id_sexo INT NOT NULL,
    CONSTRAINT fk_personas_sexo FOREIGN KEY (id_sexo) REFERENCES sexos(id_sexo)
);

```

```

CREATE INDEX idx_personas_dni ON personas(dni);

```

-- Tabla de Cuidadores

```

CREATE TABLE IF NOT EXISTS cuidadores (
    id_cuidador INT AUTO_INCREMENT PRIMARY KEY,
    id_persona INT NOT NULL,
    fecha_ingreso DATE,
    especialidad VARCHAR(100),
    CONSTRAINT fk_cuidadores_persona FOREIGN KEY (id_persona) REFERENCES personas(id_persona)
);

```

-- Tabla de Cuidadores por Habitats

```

CREATE TABLE IF NOT EXISTS cuidadores_por_habitats (
    id_cuidador INT,
    id_habitat INT,
    PRIMARY KEY (id_cuidador, id_habitat),
        CONSTRAINT fk_cuidadores_por_habitats_cuidador FOREIGN KEY (id_cuidador) REFERENCES
cuidadores(id_cuidador),
        CONSTRAINT fk_cuidadores_por_habitats_habitat FOREIGN KEY (id_habitat) REFERENCES
habitats(id_habitat)
);

```

-- Tabla de Guías

```

CREATE TABLE IF NOT EXISTS guias (

```

```
id_guia INT AUTO_INCREMENT PRIMARY KEY,
id_persona INT NOT NULL,
fecha_ingreso DATE,
especialidad VARCHAR(100),
CONSTRAINT fk_guias_persona FOREIGN KEY (id_persona) REFERENCES personas(id_persona)
);
```

-- Tabla de Recorridos

```
CREATE TABLE IF NOT EXISTS recorridos (
    id_recorrido INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    duracion_estimada INT, -- en minutos
    descripcion TEXT
);
```

```
CREATE INDEX idx_recorridos_nombre ON recorridos(nombre);
```

-- Tabla de Zonas por Recorrido

```
CREATE TABLE IF NOT EXISTS zonas_por_recorrido (
    id_recorrido INT,
    id_zona INT,
    orden INT,
    PRIMARY KEY (id_recorrido, id_zona),
    CONSTRAINT fk_zonas_por_recorrido_recorrido FOREIGN KEY (id_recorrido) REFERENCES
    recorridos(id_recorrido),
    CONSTRAINT fk_zonas_por_recorrido_zona FOREIGN KEY (id_zona) REFERENCES zonas(id_zona)
);
```

-- Tabla de Horarios de Apertura

```
CREATE TABLE IF NOT EXISTS horarios_apertura (
    id_horario INT AUTO_INCREMENT PRIMARY KEY,
    dia_semana VARCHAR(20) NOT NULL,
    horario_apertura TIME,
    horario_cierre TIME,
    id_reserva INT NOT NULL,
    CONSTRAINT fk_horarios_apertura_reserva FOREIGN KEY (id_reserva) REFERENCES
    reserva(id_reserva)
);
```

```
CREATE INDEX idx_horarios_apertura_dia_semana ON horarios_apertura(dia_semana);
```

-- Tabla de Horarios de Recorridos

```
CREATE TABLE IF NOT EXISTS horarios_recorridos (
    id_recorrido INT,
    id_guia INT,
    dia DATE,
    hora TIME,
```

```
PRIMARY KEY (id_recorrido, id_guia, dia, hora),  
    CONSTRAINT fk_horarios_recorridos_recorrido FOREIGN KEY (id_recorrido) REFERENCES  
recorridos(id_recorrido),  
    CONSTRAINT fk_horarios_recorridos_guia FOREIGN KEY (id_guia) REFERENCES guias(id_guia)  
);  
  
CREATE INDEX idx_horarios_recorridos_dia_hora ON horarios_recorridos(dia, hora);
```