



**Tecnológico Nacional de México
Instituto Tecnológico de Tijuana**

Departamento de Sistemas y Computación
Ingeniería en Sistemas Computacionales

SEMESTRE: Agosto - Diciembre 2025

Graficación - SCC-1010

Entrega Final

César Lepe Garcia - C22212360

RAY BRUNETT PARRA GALAVIZ

3 de diciembre de 2025

1. Funcionamiento del Sistema

El sistema fue desarrollado utilizando el lenguaje de programación **Python**, implementando una arquitectura de software **Modelo-Vista-Controlador (MVC)**. Esta estructura garantiza la separación de responsabilidades, facilitando la escalabilidad y el mantenimiento del código.

Como se observa en la Figura 1, el diseño funcional permite que el usuario interactúe con el catálogo de animales, visualice modelos y consulte información detallada.

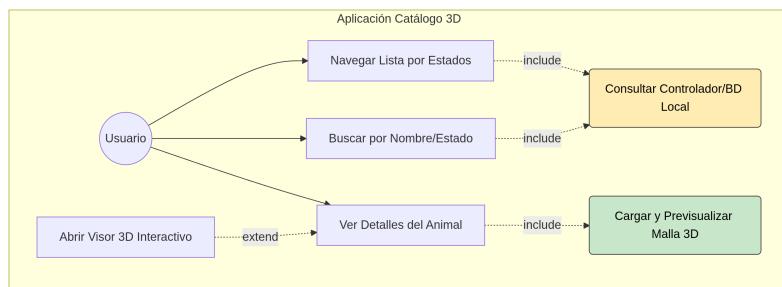


Figura 1 Diagrama de casos de uso general del sistema.

1.1. Interfaz de Usuario (Vista)

Para la capa de la Vista, se utilizó la biblioteca estándar **Tkinter** para la construcción de la interfaz gráfica de usuario (GUI), proporcionando ventanas y controles nativos. La Figura 2 muestra la ventana principal de la aplicación.

El sistema proporciona un módulo de búsqueda avanzado (Figura 3), el cual permite filtrar resultados separando por estados de conservación, nombres comunes o nombres científicos de los animales.

Una vez seleccionado un espécimen, se despliega una interfaz de detalle (Figura 4). Esta sección presenta una descripción consumida dinámicamente desde la API de Wikipedia, acompañada de una previsualización del modelo.

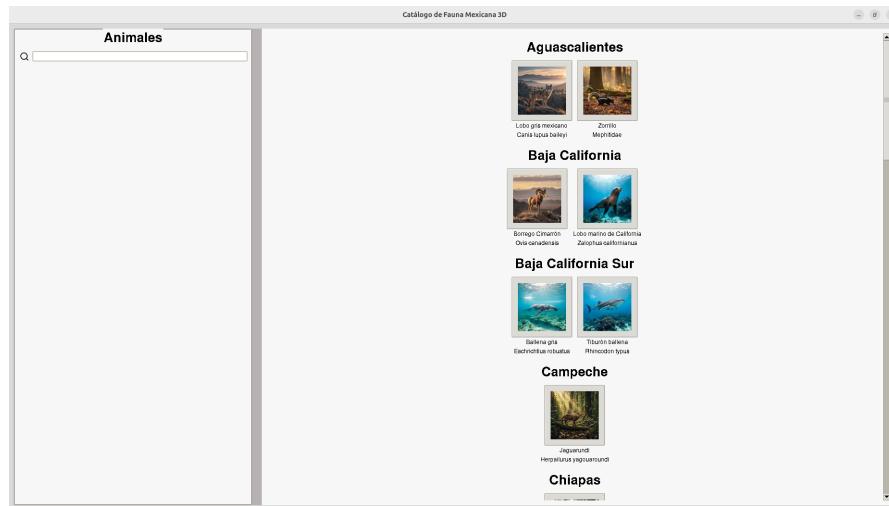


Figura 2 Ventana principal de la aplicación desarrollada en Tkinter.

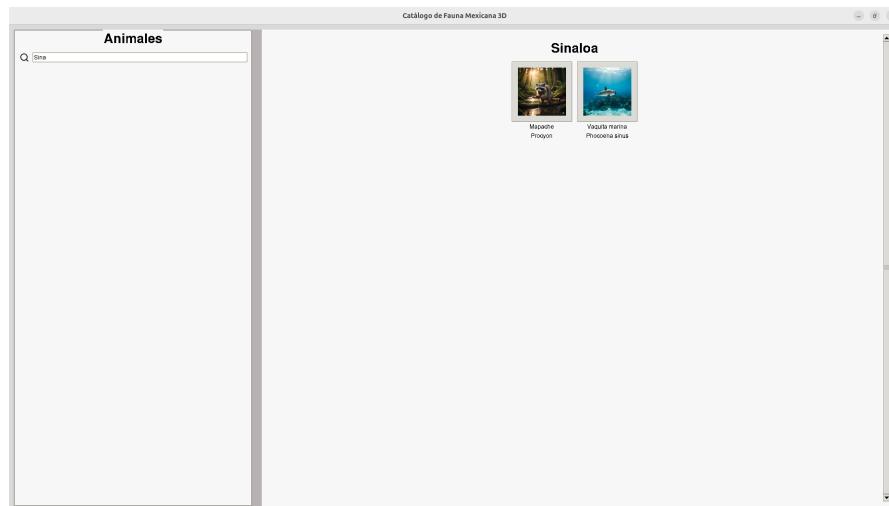


Figura 3 Módulo de búsqueda con filtros por nombre y estado.

Un componente distintivo de la interfaz es la visualización tridimensional. Para ello se integró **PyVista**, una librería que permite renderizar y manipular mallas tridimensionales en tiempo real, permitiendo al usuario rotar y examinar el modelo (Figura 5).

1.2. Gestión de Datos y Lógica (Modelo y Controlador)

Para la gestión de datos, se optó por una solución híbrida que combina almacenamiento local y consultas web:



Figura 4 Ficha técnica con descripción extraída de Wikipedia.

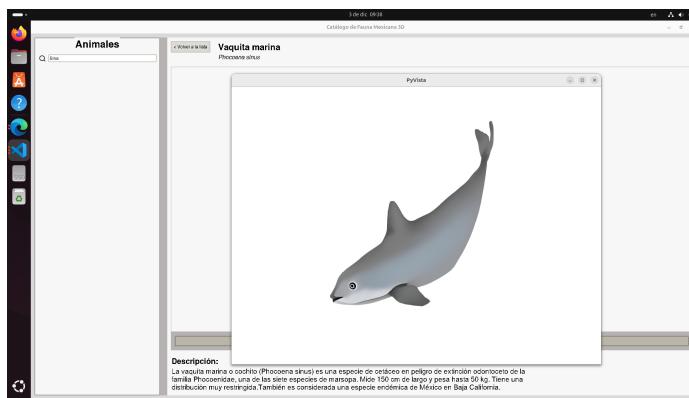


Figura 5 Visualización interactiva del modelo 3D mediante PyVista.

- **SQLite:** Actúa como base de datos local para almacenar registros estáticos, rutas de archivos y configuraciones del sistema.
- **API de Wikipedia:** Se consume este servicio para obtener descripciones dinámicas y actualizadas, reduciendo el peso de la base de datos local.

El flujo de interacción completo, desde la solicitud del usuario en la GUI hasta la recuperación de datos en la API y la base de datos, se detalla en el diagrama de secuencia de la Figura 6.

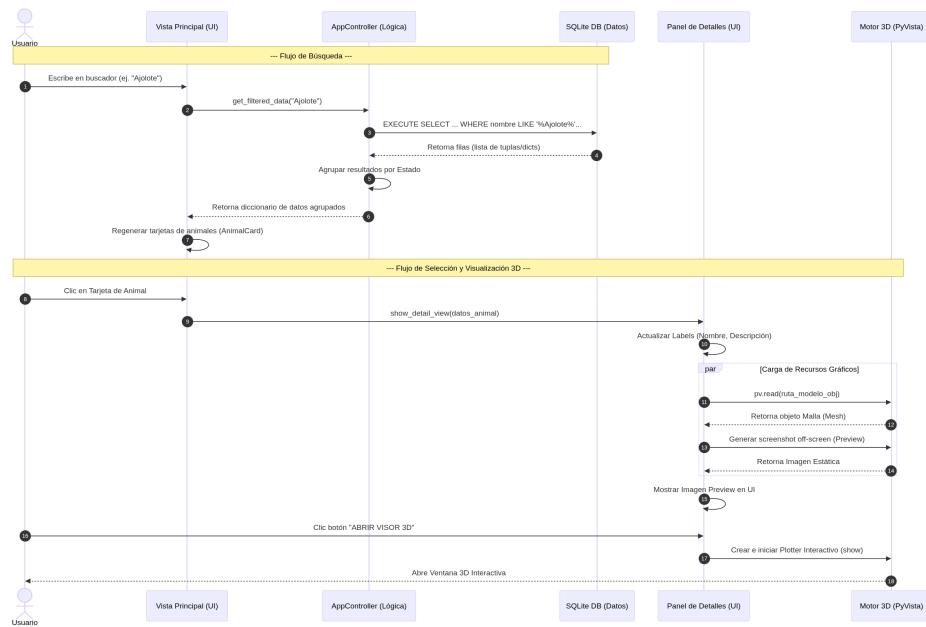


Figura 6 Diagrama de secuencia: Flujo de interacción entre GUI, Controlador y API.