Submarinismo y pesca

Felip Lloret Julià

Fecha: 22/05/2024

ÍNDICE

Introducción

Se pretende realizar una aplicación para un club de submarinismo que se encuentra en varios lugares del mediterráneo concretamente en Barcelona,Valencia y Murcia, en el cuál se harán expediciones semanales en busca de peces y distintos moluscos o crustáceos, toda animal que sean encontrados se introducirán en la base de datos por su nombre, nombre científico, lugar (ciudad) dónde ha sido encontrado y la fecha. Además habrá una función a parte con la que podremos introducir si queremos el cebo al que más recurre esta especie. Por último tendremos un botón el cuál nos permitirá ver el contenido ya introducido anteriormente en la base de datos. La aplicación realmente tendrá la función de recontar las especies que viven en estas aguas y en caso de extinción de alguna de ellas poder obtener la última fecha y lugar donde se vieron.

Herramientas y Métodos

CHATGPT,UMBRELLO,DRAW.IO,VSCODE,MongoDB,VirtualBox.

En primer lugar se va a utilizar MongoDB para crear el servidor donde se va a albergar la base de datos. VirtualBox será necesario para la virtualización de una máquina donde se encontrará el cliente. El cliente solo necesitará el vscode para interpretar el código que será generado por la inteligencia artificial (chatgpt). Draw.io se utilizará para crear la perspectiva estática, como por el ejemplo la entidad relación o el paso a tablas Umbrello en cambio se utilizará para creación del diagrama de casos de uso o el diagrama de clases.

Perspectiva estática

La perspectiva estática se centra en la estructura y los componentes de un sistema en un momento específico, sin considerar su comportamiento en el tiempo. Es decir, se enfoca en los elementos que componen el sistema y sus relaciones.

ENTIDAD / RELACIÓN Y PASO A TABLAS

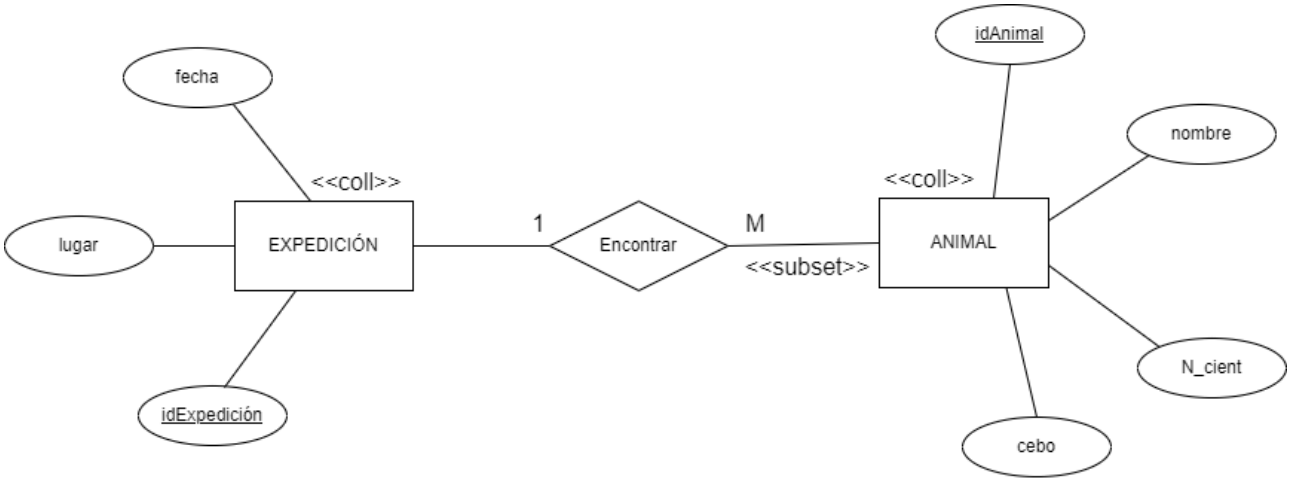

Fig.1: Al ser una relación de uno 1 a M se ha utilizado el “subset pattern” este patrón implica almacenar un subconjunto de datos en un documento para evitar consultas excesivamente grandes. La referencia de **idExpedicion** en la colección **animales** es una forma simplificada de mantener un subconjunto de la información necesaria.

DDL (Data Definition Language)

Se utiliza para definir la estructura y esquema de la base de datos. En el código, esto ocurre cuando defines las colecciones y los campos en MongoDB.

```
if 'club_submarinismo' not in client.list_database_names():
    db = client.club_submarinismo

if 'expediciones' not in db.list_collection_names():
    expediciones = db.create_collection('expediciones')

if 'animales' not in db.list_collection_names():
    animales = db.create_collection('animales')
```

DML (Data Manipulation Language)

Se utiliza para manipular los datos dentro de la base de datos. En el código, esto ocurre cuando insertas nuevos documentos en las colecciones MongoDB.

```
expedicion_id = expediciones.insert_one({'fecha': fecha, 'lugar': lugar}).inserted_id
animales.insert_one({'nombre': animal, 'N_cient': N_cient, 'idExpedicion': expedicion_id, 'cebo': None})
```

DQL (Data Query Language

Se utiliza para consultar datos dentro de la base de datos. En el código, esto ocurre cuando recuperas datos de las colecciones MongoDB.

```
animales_registrados = animales.aggregate([
    {
        '$lookup': {
            'from': 'expediciones',
            'localField': 'idExpedicion',
            'foreignField': '_id',
            'as': 'expedicion'
        }
    },
    {
        '$unwind': '$expedicion'
    }
])
```

DCL (Data Control Language)

Se utiliza para controlar el acceso de los usuarios al programa.Este fragmento asegura que solo el usuario "Felip" tenga acceso al programa y muestra un mensaje de error para los demás usuarios.

```
#usuario_actual = getpass.getuser()
#if usuario_actual != "Felip":
#    messagebox.showerror("Error de Acceso", "Lo siento, no tienes permiso para usar este programa.")
#    exit()
```

Perspectiva dinámica

La perspectiva dinámica se ocupa del comportamiento del sistema a lo largo del tiempo. Se interesa en cómo interactúan los componentes del sistema y cómo se desarrollan los procesos en el tiempo.

Sketch:

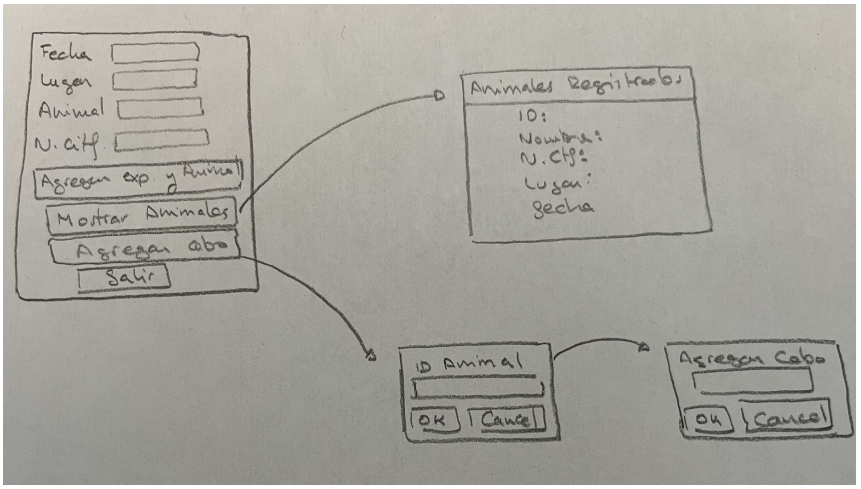

Fig.2: Boceto inicial del programa para ver, que nuevas ventanas se abrirán al interactuar con la botonera de la interfaz principal.

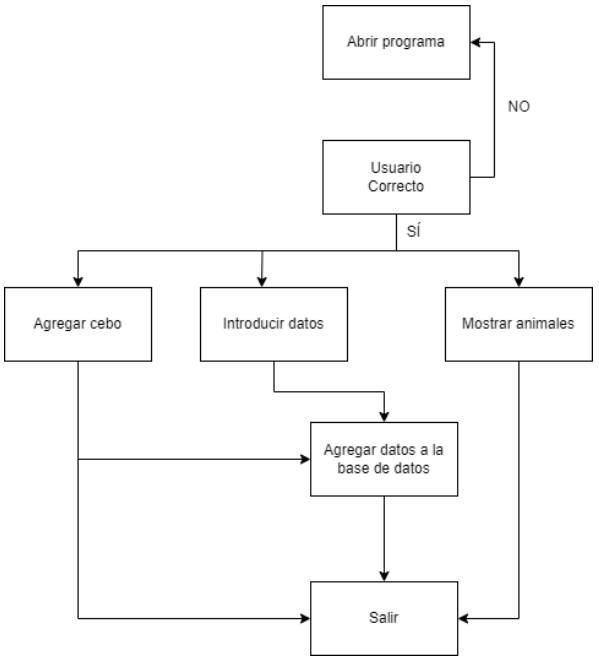

Fig.3: Algoritmo DFD para ver el funcionamiento de la aplicación.

Casos de Uso:

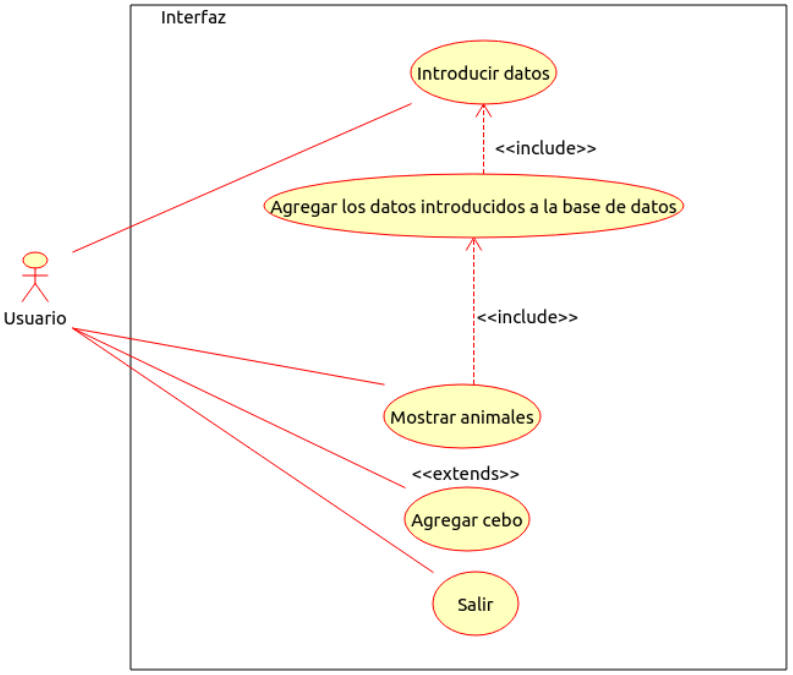

Fig.4: El diagrama de casos de uso es una representación gráfica de las interacciones entre los actores externos (usuarios o sistemas) y un sistema particular. Este tipo de diagrama se utiliza comúnmente en el análisis y diseño de sistemas para capturar y comunicar los requisitos funcionales del sistema.

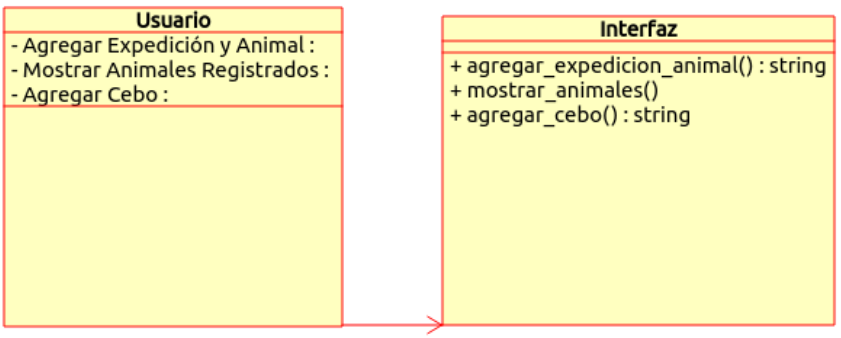

Fig.5: El diagrama de clases es una representación gráfica de los objetos del sistema y sus relaciones. Es una parte fundamental del modelado de sistemas orientados a objetos y se utiliza principalmente en la fase de diseño para visualizar la estructura estática de un sistema.

1. Agregar Expedición y Animal

Descripción: Este caso de uso permite al usuario agregar una nueva expedición junto con un animal encontrado durante esa expedición.
Actores: Usuario
Flujo principal: El usuario ingresa la fecha y el lugar de la expedición, así como el nombre y el nombre científico del animal.
El sistema verifica que todos los campos obligatorios estén completos.
El sistema verifica si el animal o nombre científico ya existe en la base de datos.
Si el animal no existe, el sistema agrega la expedición y luego el animal a la base de datos.
El sistema muestra un mensaje de éxito al usuario.

2. Mostrar Animales

Descripción: Este caso de uso permite al usuario ver la lista de animales registrados en la base de datos junto con los detalles de la expedición en la que fueron encontrados.
Actores: Usuario
Flujo principal:
El usuario selecciona la opción para mostrar los animales registrados.
El sistema recupera la información de los animales y las expediciones asociadas desde la base de datos.
El sistema muestra la lista de animales junto con sus detalles en una ventana emergente.

3. Agregar Cebo

Descripción: Este caso de uso permite al usuario agregar información sobre el cebo utilizado para atraer a un animal en particular.
Actores: Usuario
Flujo principal:
El usuario proporciona el ID del animal del cual desea agregar información sobre el cebo.
El usuario ingresa el tipo de cebo utilizado.
El sistema verifica que el ID del animal proporcionado esté registrado en la base de datos.
El sistema actualiza el registro del animal con la información del cebo.
El sistema muestra un mensaje de éxito al usuario.

Implementación:

Agregar expedición y animal:
Se implementa en la función **agregar_expedicion_animal()**, donde se recopilan los datos ingresados por el usuario, se verifican y se agregan a la base de datos.
Mostrar Animales:
Se implementa en la función **mostrar_animales()**, que realiza una consulta a la base de datos para recuperar la información de los animales y las expediciones asociadas, y luego muestra esta información al usuario en una ventana emergente.
Agregar Cebo:
Se implementa en la función **agregar_cebo()**, que permite al usuario proporcionar el ID del animal del cual desea agregar información sobre el cebo, y luego actualiza el registro del animal con la información del cebo en la base de datos.

Conclusiones

Durante el desarrollo de este proyecto, se ha logrado crear una aplicación funcional y útil para el club de submarinismo, que opera en varias ubicaciones a lo largo del Mediterráneo, específicamente en Barcelona, Valencia y Murcia. Esta aplicación ha sido diseñada para facilitar la gestión de expediciones semanales en busca de peces, moluscos y crustáceos, así como para mantener un registro detallado de los especímenes encontrados.

Una de las principales funcionalidades de la aplicación es la capacidad de introducir datos de cada especie encontrada, incluyendo su nombre común, nombre científico, ubicación (ciudad) y fecha del hallazgo. Esto permite mantener un registro histórico de las diferentes especies avistadas en cada ubicación y en qué momentos del año son más comunes.

Además, se ha implementado una función adicional que permite registrar el cebo al que más recurre cada especie. Esto puede proporcionar información valiosa para futuras expediciones, ayudando a los miembros del club a prepararse mejor y aumentar sus posibilidades de éxito en la captura de especies específicas.

La aplicación también cuenta con una función de visualización de datos, que permite a los usuarios acceder fácilmente al contenido introducido anteriormente en la base de datos. Esto facilita la consulta de información previa y el análisis de tendencias a lo largo del tiempo.

En resumen, la aplicación desarrolla una herramienta integral para la gestión de expediciones y el registro de especies marinas encontradas. Su implementación contribuirá significativamente a mejorar la eficiencia y la experiencia general de los miembros del club en sus actividades de submarinismo en el Mediterráneo.

Bibliografía y Webgrafía

draw.io - free flowchart maker and diagrams online. (s. f.). https://app.diagrams.net/

ChatGPT. (s. f.). https://chat.openai.com/

REPOSITORIO GH: https://github.com/FelipDAM/BD2