

Submarinismo y pesca

Felip Lloret Julià

Fecha: 19/04/2024

ÍNDICE

1. **Introducción**
2. **Herramientas y Métodos**
3. **Perspectiva Estática**
4. **Perspectiva Dinámica**
5. **Conclusiones**
6. **Bibliografía y Webgrafía**

Introducción

Se va a 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 sea encontrado se introducirá en la base de datos con su nombre, nombre científico, lugar (ciudad) dónde ha sido encontrado y la fecha. Además tendremos 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.

Herramientas y Métodos

CHATGPT, UMBRELLO, DRAW.IO, VSCODE, SQLITE.

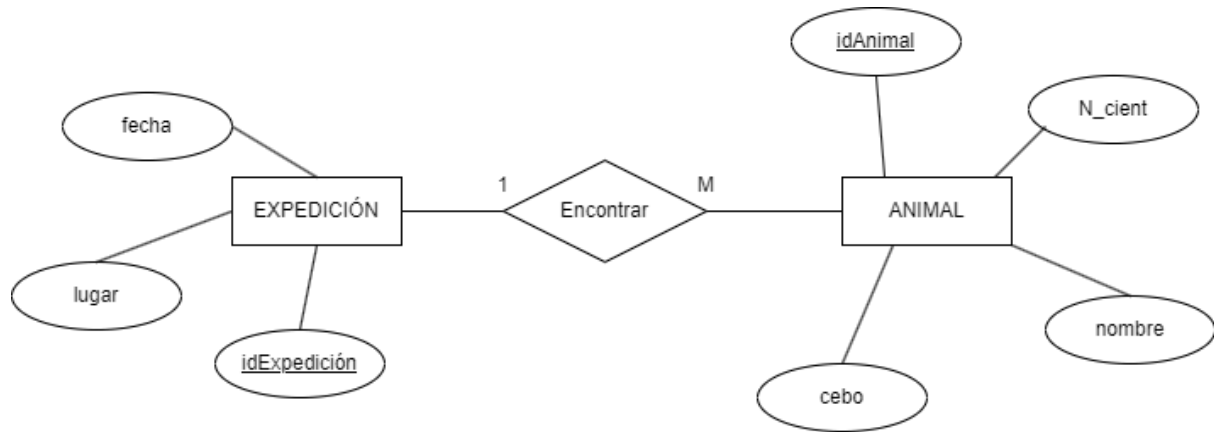
En primer lugar, se ha utilizado draw.io software de dibujo de gráficos multiplataforma para el diseño de la entidad relación y el sketch, umbrello para el diseño del diagrama de clases de uso y el diagrama de clases.

Más adelante se ha utilizado la herramienta de inteligencia artificial chatgpt para la generación del código del programa, base de datos y la conexión entre ellas.

Vscode se ha utilizado para albergar el código del programa y SQLITE para la base datos.

Perspectiva estática

ENTIDAD / RELACIÓN



PASOS A TABLAS

ANIMAL = idAnimal + N_cient + nombre + cebo + idExpedición

C.Ali: idExpedición → EXPEDICIÓN (idExpedición)

VNN: N_cient

nombre

EXPEDICIÓN = idExpedición + fecha + lugar

VNN: fecha

lugar

DDL (Data Definition Language)

El código utiliza DDL para definir la estructura de la base de datos, específicamente con las declaraciones CREATE TABLE. Estas declaraciones definen las tablas EXPEDICIÓN y ANIMAL junto con sus columnas y restricciones.

```
cursor.execute('''
    CREATE TABLE IF NOT EXISTS EXPEDICION (
        idExpedicion INTEGER PRIMARY KEY,
        fecha TEXT,
        lugar TEXT
    )
''')

cursor.execute('''
    CREATE TABLE IF NOT EXISTS ANIMAL (
        idAnimal INTEGER PRIMARY KEY,
        nombre TEXT,
        N_cient TEXT,
        idExpedicion INTEGER,
        cebo TEXT,
        FOREIGN KEY (idExpedicion) REFERENCES EXPEDICION(idExpedicion),
        CONSTRAINT unique_nombre UNIQUE (nombre)
    )
''')
```

DML (Data Manipulation Language)

El código utiliza DML para manipular los datos en la base de datos, principalmente con las declaraciones INSERT INTO. Estas declaraciones agregan nuevos registros a las tablas EXPEDICIÓN y ANIMAL.

```
cursor.execute('INSERT INTO EXPEDICION (fecha, lugar) VALUES (?, ?)',
(fecha, lugar))
conn.commit()

id_expedicion = cursor.lastrowid

cursor.execute('INSERT INTO ANIMAL (nombre, N_cient, idExpedicion)
VALUES (?, ?, ?)', (animal, N_cient, id_expedicion))
conn.commit()
```

DQL (Data Query Language)

El código utiliza DQL para consultar la base de datos, específicamente con la declaración SELECT. Esta declaración se utiliza para recuperar información de la base de datos, como la lista de animales registrados durante las expediciones.

```
cursor.execute('''
        SELECT ANIMAL.idAnimal, ANIMAL.nombre, ANIMAL.N_cient,
        ANIMAL.cebo, EXPEDICION.lugar, EXPEDICION.fecha
        FROM ANIMAL
        LEFT JOIN EXPEDICION ON ANIMAL.idExpedicion =
        EXPEDICION.idExpedicion
    ''')
animales = cursor.fetchall()
```

DCL (Data Control Language)

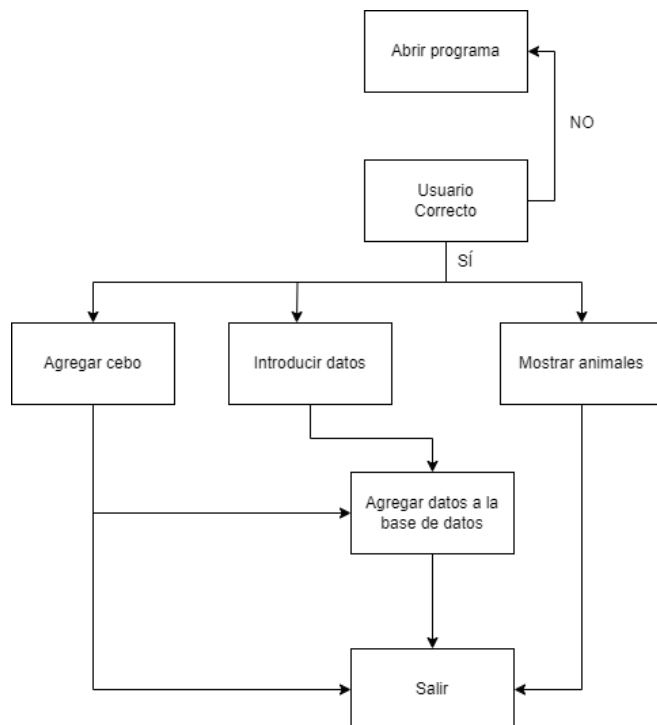
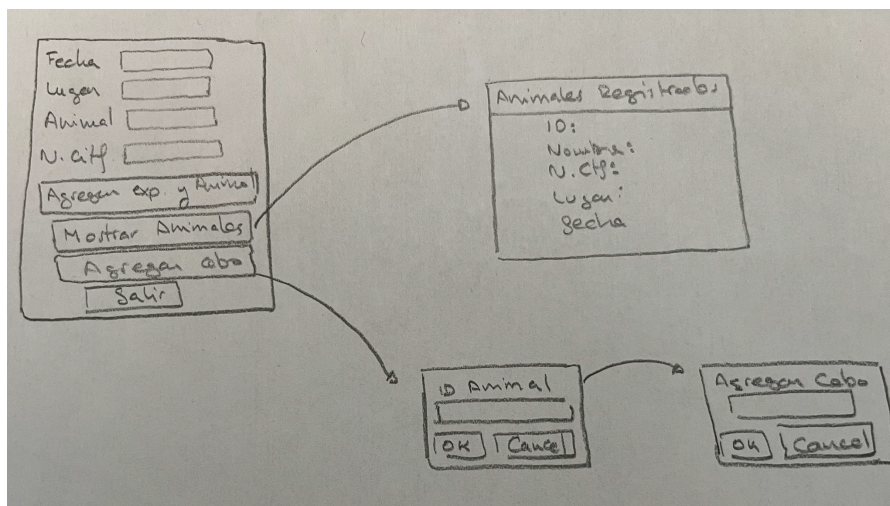
Se está verificando si el usuario actual no es "Felip". Si esta condición es verdadera, se muestra un mensaje de error y el programa se cierra, lo que refleja un control de acceso discrecional sobre quién puede ejecutar el programa.

```
# Obtener el nombre de usuario del sistema operativo
usuario_actual = getpass.getuser()

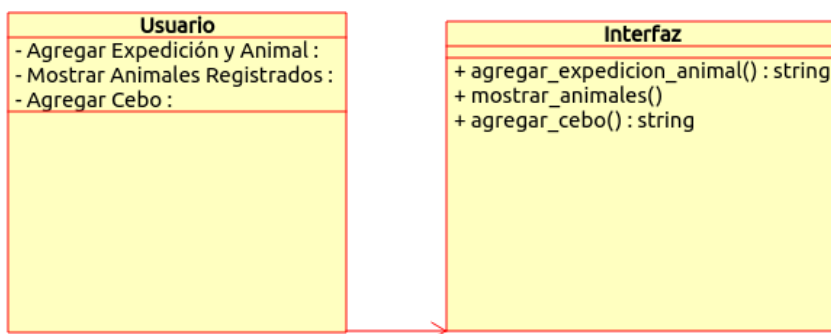
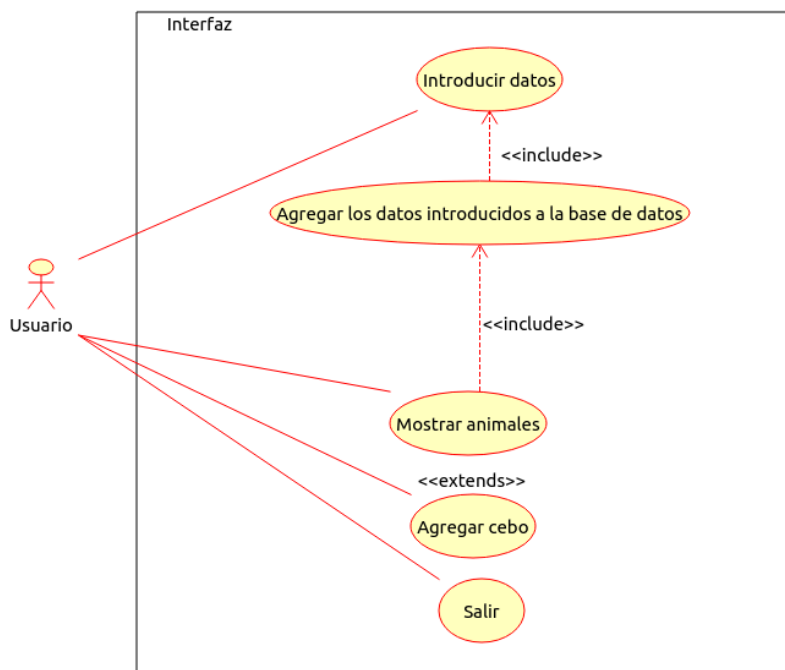
if usuario_actual != "Felip":
    messagebox.showerror("Error de Acceso", "Lo siento, no tienes
    permiso para usar este programa.")
    exit()
```

Perspectiva dinámica

Sketch:



Casos de Uso:



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/BD>