

# **IES ARROYO HARNINA**

Técnico Superior en Desarrollo de Aplicaciones Multiplataformas

PROYECTO

**FishCatch**

**Autor:** Bartolomé García Rodríguez

**Curso Académico:** 2º DAM

## **ÍNDICE**

1. INTRODUCCIÓN .....	3
2. OBJETIVOS .....	4
3. MARCO TEORICO.....	5
4. DESARROLLO DEL PROYECTO .....	6
4.1 Diseño .....	6
4.2 Implementación.....	8
5. CONCLUSIONES .....	11
BIBLIOGRAFÍA .....	11
ANEXO A: MANUAL DE USUARIO .....	12

## **1. INTRODUCCIÓN**

El proyecto FishCatch se centra en el desarrollo de una aplicación móvil para pescadores, para así, tener un diario personal en el que registrar las capturas de pesca, tanto, así como registro de especies, el tamaño, el peso y fecha. Además, contará con funcionalidades como la ubicación geolocalizada para marcar la ubicación exacta de la captura y condiciones ambientales en el momento de la captura.

## **2. OBJETIVOS**

El objetivo de la aplicación es proporcionar a los pescadores una herramienta para guardar sus capturas de pesca, para así, tener todo lo relacionado con este deporte en una sola aplicación y por tanto ser una herramienta fácil e intuitiva de utilizar.

### **3. MARCO TEORICO**

La aplicación, va a utilizar los siguientes elementos y tecnologías:

- La plataforma principal para el desarrollo de la aplicación es Android Studio. Con ella se diseñarán la interfaz de usuario, gestionaremos la Base de Datos y programaremos todas las funcionalidades necesarias en Java.
- Base de Datos Local (SQLite). Para el almacenamiento de los datos de capturas, especies, ubicaciones y condiciones ambientales, utilizaremos una Base de Datos local SQLite.
- Geolocalización (GPS). Para registrar la ubicación exacta de las capturas de pesca, utilizaremos el servicio de GPS integrado en los dispositivos Android. Mediante el uso de la API de Google Maps, la aplicación podrá mostrar en tiempo real las coordenadas geográficas del lugar de la captura y por tanto guardar dicho dato.
- OpenWeatherMap. Para integrar las condiciones ambientales en tiempo real, como la temperatura, utilizaremos la API de OpenWeatherMap.
- Análisis de datos y estadísticas. Tendremos una Activity, donde se mostrarán datos estadísticos como total de capturas, la captura con mayor peso (con y sin ubicación) y el día con mayor número de capturas.
- Permisos de Android. Para permitir que el Usuario capture imágenes o seleccione fotos desde la galería, gestionaremos los permisos de Cámara y Almacenamiento correspondientes.
- Se ha optado por integrar OSMDROID, una biblioteca de código abierto que permite la visualización de mapas basados en OpenStreetMap. Esta herramienta ofrece funcionalidades similares a Google Maps para mostrar mapas, marcadores y rutas en tiempo real, permitiendo al usuario visualizar de forma precisa las ubicaciones de sus capturas. Osmroid proporciona además soporte para geolocalización en tiempo real y permite la interacción directa con los mapas sin necesidad de depender de servicios comerciales o licencias. Gracias a esta integración, el pescador puede identificar fácilmente los puntos donde ha tenido éxito y acceder a información geoespacial sin restricciones.

## 4. DESARROLLO DEL PROYECTO

### 4.1 DISEÑO

- **MAIN ACTIVITY**

#### **Back-end:**

Se diseñó una base de datos SQLite con un esquema que incluye las tablas Especie, Captura, Ubicacion y Condiciones, estableciendo relaciones mediante claves foráneas para garantizar la integridad de los datos. La gestión de la base de datos se implementó con un patrón Singleton para optimizar el acceso y evitar múltiples instancias.

#### **Front-end**

La interfaz principal está compuesta por botones que permiten al usuario navegar entre funcionalidades: agregar capturas, visualizar capturas, consultar estadísticas y enviar sugerencias.

- **MAIN ACTIVITY AGREGAR CAPTURA**

#### **Back-end**

Se utiliza el adaptador de base de datos para almacenar la información de las capturas, incluyendo detalles como especie, peso, tamaño, fecha, hora, ubicación, temperatura y foto. La clase también integra llamadas a una API externa (OpenWeatherMap) mediante Retrofit para obtener la temperatura actual según la ubicación.

#### **Front-end**

La interfaz cuenta con múltiples controles para facilitar la entrada de datos: un spinner para seleccionar la especie, botones para añadir imagen, obtener ubicación GPS, obtener temperatura, y guardar la captura. Los campos para peso, tamaño, fecha/hora y comentarios permiten al usuario completar los detalles. Se usan diálogos para seleccionar fecha y hora, y la imagen se carga desde la galería.

- **MAIN ACTIVITY VER CAPTURAS**

#### **Back-end**

Se utiliza un adaptador de base de datos para obtener las fechas únicas en las que se registraron capturas y recuperar las capturas correspondientes a la fecha seleccionada. La clase también permite actualizar los datos tras operaciones como eliminar registros.

### **Front-end**

La interfaz incluye un Spinner para mostrar las fechas disponibles y un RecyclerView para listar las capturas correspondientes a la fecha seleccionada. Se configura un adaptador personalizado para mostrar la información de las capturas. Se muestran mensajes Toast cuando no hay capturas o fechas disponibles.

- **MAIN ACTIVITY DETALLE CAPTURA**

### **Back-end**

Se utiliza un adaptador de base de datos para obtener los detalles completos de una captura mediante su ID y para actualizar la información, como comentarios y la imagen asociada. La lógica incluye manejo de URI persistentes para imágenes seleccionadas por el usuario, asegurando que la referencia se mantenga tras reinicios.

### **Front-end**

La interfaz muestra los detalles de una captura: imagen, especie, peso, tamaño, fecha, hora, comentarios y temperatura. Permite seleccionar una nueva imagen desde el almacenamiento del dispositivo con permisos persistentes. Incluye un mapa con la ubicación exacta de la captura usando OSMdroid, mostrando un marcador. Se utiliza un botón para guardar cambios si se modifica la imagen o los comentarios adicionales.

- **MAIN ACTIVITY ESTADÍSTICAS PERSONALES**

### **Back-end**

La actividad obtiene datos estadísticos de las capturas desde la base de datos a través del adaptador AdaptadorBaseDeDatos. Se consultan datos como el total de capturas, especie más capturada, capturas por especie, día más exitoso y la captura con mayor peso con y sin ubicación, todo filtrado por año seleccionado o global. Se implementa la generación de un documento PDF con estas estadísticas usando PdfDocument para guardar en el almacenamiento público.

### **Front-end**

La interfaz presenta estadísticas personalizadas mediante TextView y un Spinner para filtrar por año. Un botón abre la ubicación de la captura con mayor peso en Google Maps si existe ubicación disponible. Otro botón genera un PDF con las estadísticas actuales, previa confirmación del usuario mediante diálogo.

## 4.2 IMPLEMENTACIÓN

- **MAIN ACTIVITY**

### **Back-end**

Se implementó la clase BaseDeDatosSingleton para crear y manejar la base de datos, crear las tablas y precargar especies. Además, la clase AdaptadorBaseDeDatos facilita las operaciones CRUD (crear, leer, actualizar y eliminar) sobre las tablas.

### **Front-end**

En MainActivity se programaron los botones para lanzar las actividades correspondientes mediante Intents. También se incluyó un diálogo para la entrada de texto, que permite enviar sugerencias por correo electrónico usando la app de email del dispositivo.

- **MAIN ACTIVITY AGREGAR CAPTURA**

### **Back-end**

Se implementa el almacenamiento de la captura en la base de datos junto con su ubicación y condiciones (temperatura). Se realiza la integración con el servicio de ubicación de Google para obtener coordenadas GPS, y con Retrofit para utilizar la API meteorológica. La clase controla permisos para acceso a ubicación y maneja la obtención y guardado de imágenes en almacenamiento local.

### **Front-end**

Se configuran eventos para todos los botones y campos interactivos: abrir la galería para seleccionar imagen, mostrar DatePicker y TimePicker para fecha/hora, solicitar permisos y obtener ubicación, y mostrar temperatura tras consultar la API. Al guardar, se validan los datos obligatorios y se notifica al usuario con mensajes Toast.

- **MAIN ACTIVITY VER CAPTURAS**

### **Back-end**

Se implementa la lógica para consultar fechas únicas y capturas por fecha desde la base de datos mediante el adaptador. También se provee un método para actualizar el spinner y la lista tras cambios en los datos (por ejemplo, al borrar una captura).



### **Front-end**

Se configuran eventos para el spinner que actualizan la lista de capturas mostrada en el RecyclerView al seleccionar una fecha. En el ciclo de vida onResume se refrescan los datos para mantener la vista actualizada. Se manejan casos en que no hay datos con mensajes.

- **MAIN ACTIVITY DETALLE CAPTURA**

### **Back-end**

Se implementa la consulta para cargar detalles de la captura por ID, manejo de URI para la imagen seleccionada con permisos de lectura persistentes, y actualización de la captura en la base de datos con los datos modificados por el usuario.

### **Front-end**

Se configuran vistas para mostrar todos los campos de detalle y se ocultan elementos no aplicables (por ejemplo, la temperatura si no está registrada). Se implementa el lanzador de actividad para seleccionar una imagen y actualizar la vista. Se integra un mapa dinámico con OSMdroid que solo aparece si la captura tiene ubicación registrada. Finalmente, un botón permite guardar los cambios y confirma con un mensaje.

- **MAIN ACTIVITY ESTADÍSTICAS PERSONALES**

### **Back-end**

Se implementan métodos para cargar años disponibles, consultas específicas filtradas por año y actualización de vistas con los resultados. La generación del PDF incluye escritura texto, con manejo de excepciones para almacenamiento en la carpeta pública Downloads. Se garantiza que la generación se confirme antes con un diálogo de alerta.

### **Front-end**

Se configuran eventos para la selección del año en el Spinner que disparan actualización automática de estadísticas. El botón para abrir Google Maps construye la URI con coordenadas y verifica disponibilidad de la app. El botón para PDF lanza un diálogo de confirmación y luego genera el archivo. Se habilitan o deshabilitan botones según la disponibilidad de datos para evitar acciones inválidas.

- **WHEATER API SERVICE**

Se define una interfaz REST usando Retrofit para consumir la API del clima de OpenWeatherMap. La interfaz contiene un método `getCurrentWeather` que realiza una petición GET al endpoint "data/2.5/weather". Esta petición recibe como parámetros la latitud, longitud, clave API (appid) y unidades de medida (por ejemplo, "metric"). El resultado se encapsula en un objeto `Call<WeatherResponse>`, que es la respuesta JSON mapeada a la clase modelo `WeatherResponse`.

- **WEATHER RESPONSE**

Esta clase representa la estructura del JSON que se recibe desde la API del clima (OpenWeatherMap). Contiene una clase anidada `Main` que tiene el campo `temp` para almacenar la temperatura actual. Esta estructura permite acceder fácilmente a la temperatura actual con `weatherResponse.main.temp` tras deserializar la respuesta JSON.

- **CAPTURA ADAPTER**

**Back-end**

Mantiene y administra la lista de objetos `Captura`.

Al enlazar cada ítem, obtiene y muestra datos relevantes: nombre de la especie, peso y foto (si existe) e implementa lógica para eliminar capturas, tanto en la lista como en la base de datos.

Además, controla excepciones al cargar imágenes para evitar que la app falle si la imagen no está disponible.

**Front-end**

Infla el layout personalizado `item_captura.xml` para cada elemento del `RecyclerView`, mostrando la imagen en miniatura, el nombre de la especie y el peso.

Define la interacción táctil en cada captura:

Clic simple abre la actividad detalle de la captura, pasando el ID por Intent.

Clic largo abre un diálogo de confirmación para eliminar la captura.

Actualiza la interfaz después de eliminar un ítem, notificando los cambios al `RecyclerView` y refrescando otros componentes relacionados en la actividad padre (como un spinner).

## 5. CONCLUSIONES

Con esta app se pretende tener organizadas todas las capturas de pesca en un solo lugar, para así, no tener que buscar datos de estas en la galería, notas...

Ha surgido viendo la necesidad de tener un diario donde tener todo organizado de forma clara y concisa, a través de la observación y preguntas a un gran número de pescadores.

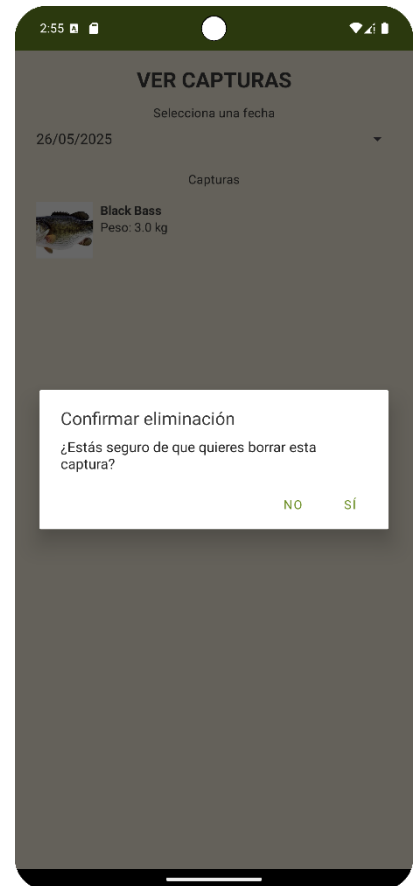
## BIBLIOGRAFÍA

### *Direcciones Web.*

- OpenWeatherMap API: <https://openweathermap.org/api>
- Retrofit: <https://square.github.io/retrofit/>
- Android Developers Documentation: <https://developer.android.com/>
- Osmdroid: Android Maps Library: <https://github.com/osmdroid/osmdroid>

## ANEXO A: MANUAL DE USUARIO

- En Ver Capturas, si hacemos un click largo sobre una de las capturas, aparecerá el dialogo para poder eliminar dicha captura:



- El PDF generado con las estadísticas personales, se guarda en la carpeta: DOWNLOADS

