

IES MONTE NARANCO

**Ciclo Formativo de Grado Superior en Desarrollo de
Aplicaciones Multiplataforma (DAM)**

GUÍA DE PESCA INTERACTIVA ASTURIANA

**Pelayo González
Curso 2024 – 2025
Oviedo**

INDICE

1. INTRODUCCIÓN

- 1.1. Contexto
- 1.2. Presentación
- 1.3. Objetivos

2. ANÁLISIS Y DISEÑO DE LA APLICACIÓN

- 2.1. Análisis de requisitos funcionales
 - 2.1.1. Introducción
 - 2.1.2. Tabla numerada de requisitos
- 2.2. Diagrama de casos de uso
 - 2.2.1. Introducción
 - 2.2.2. Diagramas y descripción
- 2.3. Diseño de interfaces
 - 2.3.1. Introducción
 - 2.3.2. Diseño de pantallas
- 2.4. Diagrama de clases
 - 2.4.1. Introducción
 - 2.4.2. Diagramas y descripción
- 2.5. Análisis de las estructuras de datos utilizadas (bases de datos, ficheros...)
 - 2.5.1. Introducción
 - 2.5.2. Diagramas y descripción
- 2.6. Fase de pruebas
 - 2.6.1. Pruebas de validación de código
 - 2.6.2. Pruebas de implantación en distintos soportes

3. IMPLEMENTACIÓN

- 3.1. Tecnologías utilizadas en el desarrollo del proyecto

4. PUESTA EN MARCHA. DESPLIEGUE

5. CONCLUSIÓN

- 5.1. Valoración personal del trabajo realizado.
- 5.2. Posibles ampliaciones

6. BIBLIOGRAFÍA Y WEBGRAFÍA

7. ANEXOS

- I – Imágenes
- II – Código completo del proyecto

1. Introducción

1.1 Contexto

La pesca recreativa y profesional es una actividad presente día a día de muchas personas. Sin embargo, el acceso a información actualizada y verídica sobre normativas, medidas legales por especie, y la posibilidad de registrar actividad propia es escasa o poco accesible en plataformas digitales. Esto genera barreras tanto para pescadores principiantes como experimentados, que carecen de una herramienta centralizada, moderna y móvil que sirva de apoyo en sus salidas al medio natural.

En este contexto, surge la necesidad de desarrollar una aplicación digital que funcione como una guía interactiva, capaz de reunir información útil sobre la pesca y permitir a los usuarios interactuar con ella de forma personalizada, segura y adaptada a las condiciones del entorno.

1.2 Presentación

El presente proyecto consiste en el desarrollo de una aplicación móvil multiplataforma, denominada **Guía de Pesca Interactiva Asturiana**, cuyo objetivo principal es ofrecer a los pescadores una herramienta moderna, accesible y práctica para consultar información clave, registrar su actividad pesquera en cualquier lugar con acceso a internet.

1.3 Objetivos

Los principales objetivos del proyecto son los siguientes:

- Desarrollar una aplicación móvil intuitiva y útil para poder usar mientras el usuario está practicando la pesca.
- Centralizar la información sobre normativas de pesca, medidas mínimas y especies reguladas.
- Permitir a los usuarios registrar su actividad pesquera, incluyendo puntos geográficos, descripciones, imágenes y fechas.
- Incorporar autenticación segura mediante JWT y HS256.
- Integrar roles diferenciados para usuarios y administradores, con funciones exclusivas de gestión.
- Habilitar funcionalidades como recuperación de contraseña por correo electrónico y visualización del clima mediante API externa.
- Garantizar la compatibilidad y despliegue sobre servidores.

2. ANÁLISIS Y DISEÑO DE LA APLICACIÓN

2.1 Análisis y requisitos funcionales

2.1.1 Introducción

Aquí definiremos funcionalidades que debe cumplir el sistema desde el punto de vista del usuario final. Se han tenido en cuenta las necesidades tanto de los usuarios generales (pescadores) como de los administradores encargados de la gestión de contenido.

2.1.2 Tabla numerada de requisitos

Número de requisito	RF1
Nombre de requisito	Login
Descripción	El usuario podrá acceder a la app introduciendo su correo electrónico y contraseña en una pantalla de inicio de sesión.
Casos de uso asociado	Login

Número de requisito	RF2
Nombre de requisito	Registro del usuario
Descripción	El usuario podrá darse de alta en la aplicación introduciendo sus datos personales y una contraseña personal
Casos de uso asociado	Registro

Número de requisito	RF3
Nombre de requisito	Recuperación de contraseña
Descripción	El usuario podrá pedir una contraseña introduciendo su correo electrónico
Casos de uso asociado	Recuperación contraseña

Número de requisito	RF4
Nombre de requisito	Obtener el tiempo
Descripción	El usuario podrá buscar el tiempo mediante un buscado de una ciudad en del territorio español.
Casos de uso asociado	Obtener el tiempo

Número de requisito	RF5
Nombre de requisito	Visualización del mapa
Descripción	La aplicación mostrará un mapa interactivo con los puntos de pesca registrados, permitiendo al usuario explorarlos y seleccionar ubicaciones.
Casos de uso asociado	Visualizar mapa

Número de requisito	RF6
Nombre de requisito	Registro de punto de pesca
Descripción	El usuario podrá guardar un nuevo punto geográfico en el mapa incluyendo su ubicación, nombre y descripción.
Casos de uso asociado	Guardar punto

Número de requisito	RF7
Nombre de requisito	Modificación de un punto de pesca
Descripción	El usuario podrá modificar los valores variables de un punto registrado, excepto la latitud y longitud.
Casos de uso asociado	Modificar punto

Número de requisito	RF8
Nombre de requisito	Eliminación de un punto de pesca
Descripción	El usuario podrá eliminar un punto de pesca
Casos de uso asociado	Eliminar punto

Número de requisito	RF9
Nombre de requisito	Registro de captura
Descripción	El usuario podrá registrar un nuevo registro de una captura, indicando fecha, descripción e imágenes.
Casos de uso asociado	Registro de captura

Número de requisito	RF10
Nombre de requisito	Modificación de captura
Descripción	El usuario podrá modificar los valores de un registro.
Casos de uso asociado	Modificación de captura

Número de requisito	RF11
Nombre de requisito	Eliminar captura
Descripción	El usuario podrá eliminar una captura
Casos de uso asociado	Eliminar captura

Número de requisito	RF12
Nombre de requisito	Consulta de medidas
Descripción	El usuario podrá consultar y filtrar las medidas mínimas impuestas por el principado de cada especie.
Casos de uso asociado	Consulta de medidas

Número de requisito	RF13
Nombre de requisito	Perfil de usuario
Descripción	El usuario podrá consultar sus datos personales
Casos de uso asociado	Ver perfil

Número de requisito	RF14
Nombre de requisito	Modificar perfil de usuario
Descripción	El usuario podrá modificar sus datos personales y su contraseña
Casos de uso asociado	Modificar perfil de usuario

Número de requisito	RF15
Nombre de requisito	Gestión de usuarios
Descripción	El administrador tendrá habilitadas funciones para habilitar o deshabilitar a los usuarios, estos no podrán hacer acceder a la aplicación.
Casos de uso asociado	Gestión usuarios

Número de requisito	RF16
Nombre de requisito	Crear Medidas
Descripción	El administrador podrá crear una nueva medida de una especie.
Casos de uso asociado	Creación de medidas

Número de requisito	RF17
Nombre de requisito	Modificación de medidas
Descripción	El administrador podrá modificar los valores de una medida.
Casos de uso asociado	Modificación de medidas

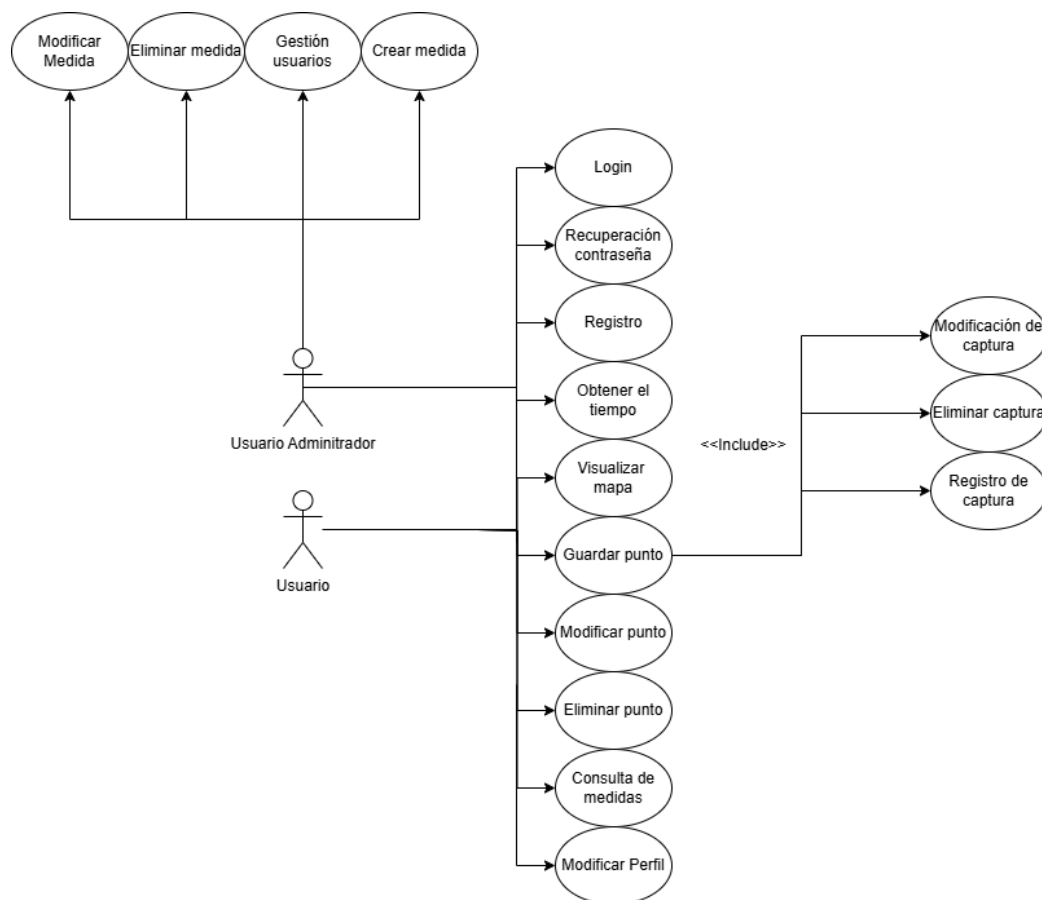
2.2. Diagrama de casos de uso

2.2.1 Introducción

El siguiente diagrama de casos de uso representa las principales funcionalidades del sistema y las interacciones que pueden realizar los distintos tipos de usuarios: usuarios y administradores.

Para las funcionalidades comunes se necesitará hacer un login y un token, para las funcionalidades de gestión se deberá tener el rol de administrador

2.2.2 Diagramas y descripción



2.3 Diseño de interfaces

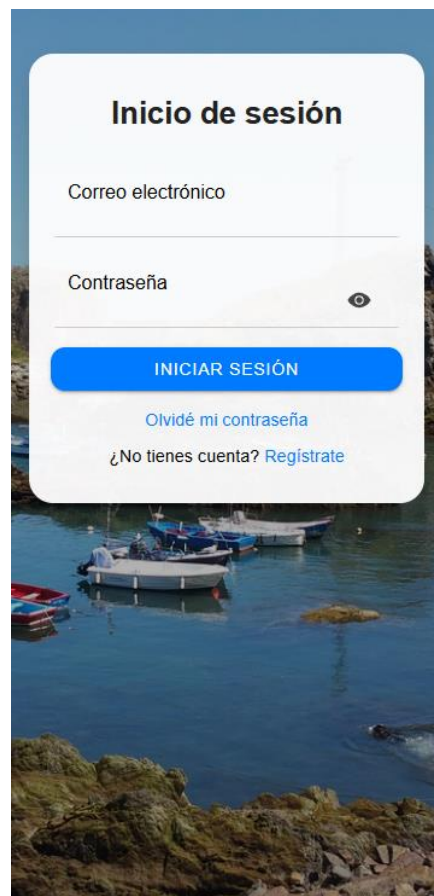
2.3.1 Introducción

Detallaremos los diseños de cada pantalla de la aplicación

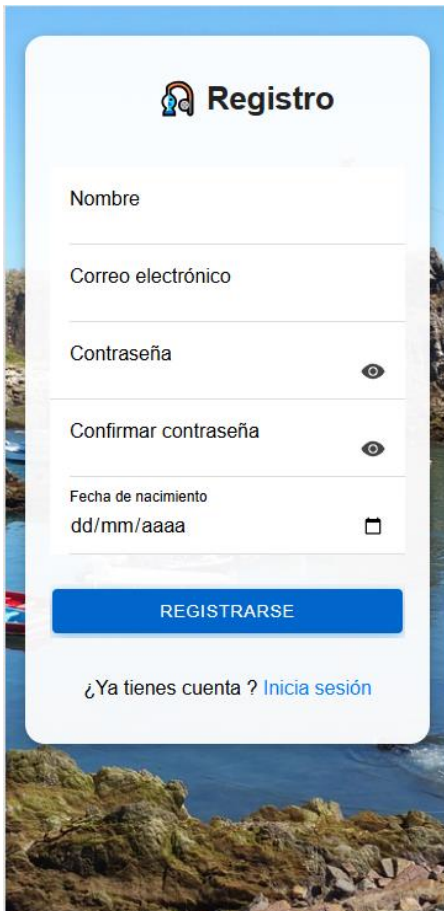
2.3.2 Diseño de interfaces



Inicio de aplicación



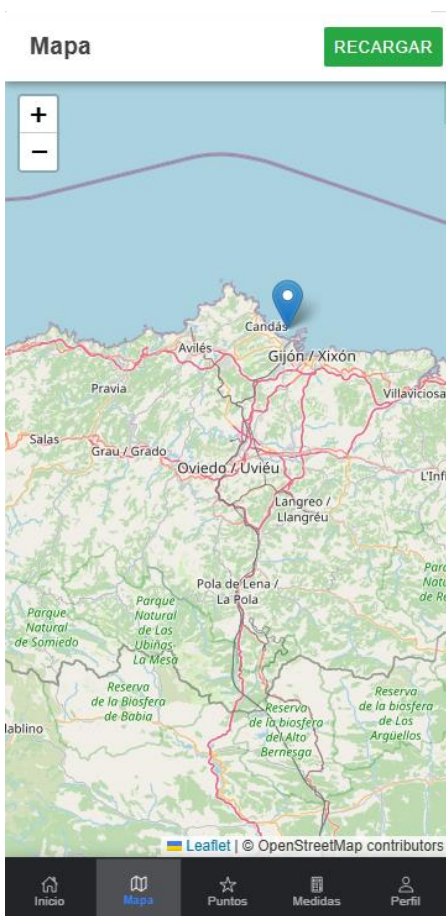
Login

The 'Registro' screen features a white registration form on a blue background with a coastal scene. The form includes fields for 'Nombre', 'Correo electrónico', 'Contraseña' (with an eye icon), 'Confirmar contraseña' (with an eye icon), and 'Fecha de nacimiento' (with a calendar icon). A blue 'REGISTRARSE' button is at the bottom, followed by a link '¿Ya tienes cuenta? Inicia sesión'.

Registro

The 'Consulta del Tiempo' screen has a blue header and a white weather card. It shows the city 'Gijón' in a search bar, a 'BUSCAR TIEMPO' button, and current weather for 'Gijón' at '21°C' with a cloud icon. Below, it lists 'Sensación térmica: 21°C', 'Muy Nuboso', 'Humedad: 78%', 'Presión: 1013 hPa', 'Viento: 0.45 m/s Este (103°)', 'Nubosidad: 61%', and 'Ráfagas de viento: 0.45 m/s'. A bottom navigation bar includes 'Inicio', 'Mapa', 'Puntos', 'Medidas', and 'Perfil'.

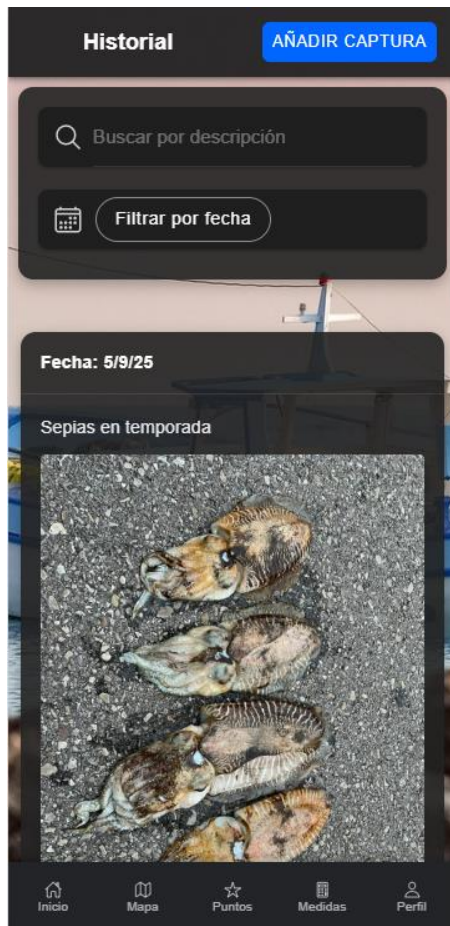
Inicioe usuario



Mapa

The 'Mis puntos' screen shows a list of fishing spots. The first entry is 'Ribadesella' with coordinates 'Latitud: 43.50274', 'Longitud: -5.05234', and description 'Descripción: prueba'. It has buttons 'EDITAR', 'HISTORIAL', and 'ELIMINAR'. The second entry is 'Secreto' with coordinates 'Latitud: 43.58362', 'Longitud: -5.71907', and description 'Descripción: Pesca de cefalópodos', also with 'EDITAR', 'HISTORIAL', and 'ELIMINAR' buttons. A bottom navigation bar is at the bottom.

Puntos



Historial



Medidas



Perfil



Gestión usuarios

2.4.1 Introducción

Diagrama de clases de la API



2.5 Análisis de las estructuras de datos utilizadas

2.5.1 Introducción

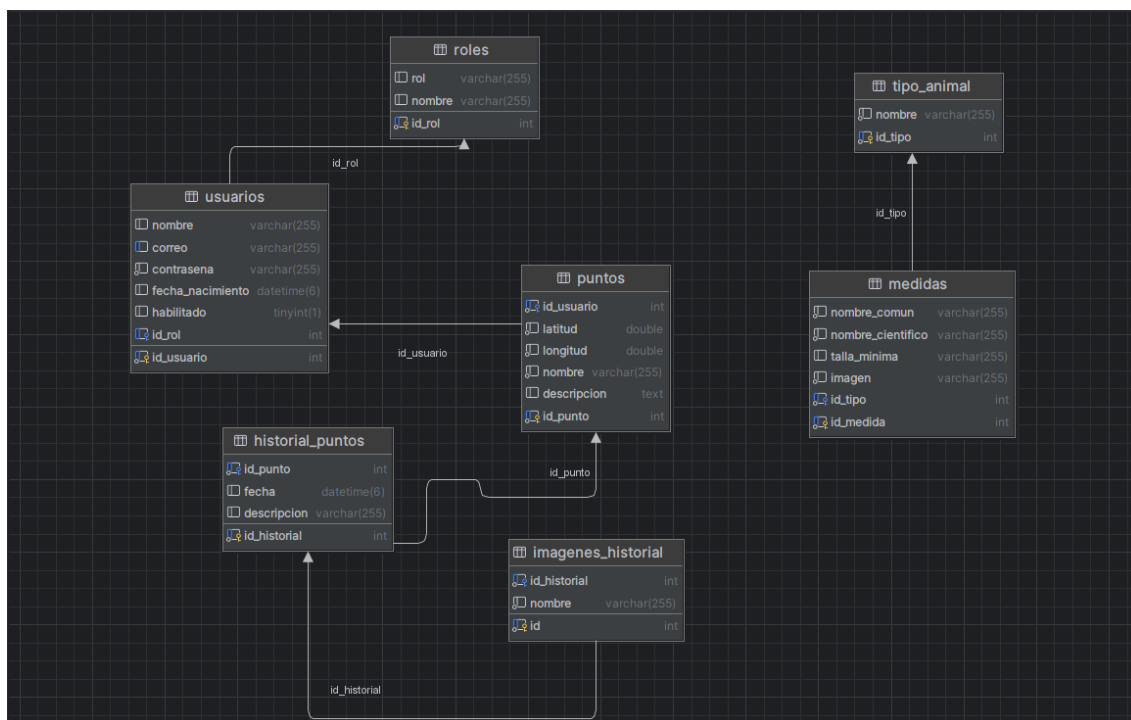
En esta sección explicaremos las estructuras que hemos desarrollado en las distintas partes que componen la aplicación

2.5.2 Diagramas y descripción

Diagrama de la base de datos

Se desarrolló con el lenguaje MYSQL, en el cual la aplicación tiene un usuario que puede tener roles, cada usuario tiene una lista de puntos que a su vez tiene una lista de historiales, que estos tienen una lista de imágenes que representan las capturas.

Cualquier usuario puede consultar la tabla medidas compuesta por tipos de animales, estos donde podremos visualizar distintas especies con los tamaños mínimos.



Almacenamiento y gestión de imágenes

Imágenes estáticas (frontend)

Las imágenes de fondo utilizadas para ambientar la interfaz gráfica (por ejemplo, fondo del login, mapa o perfil) se almacenan directamente en el proyecto frontend, dentro de la carpeta:

`src/assets/img/fondos/`

Estas imágenes están predefinidas y se cargan localmente sin necesidad de conexión al servidor. Se utilizan para mejorar la experiencia visual del usuario.

Imágenes dinámicas (backend)

Las imágenes correspondientes a capturas de puntos (como fotografías tomadas en el terreno) o imágenes de medidas se almacenan en el servidor, siguiendo esta lógica:

Las imágenes no se guardan como blobs en la base de datos, sino como archivos físicos en el sistema de archivos del servidor, por ejemplo:

`/images/capturas/`

`/images/medidas/`

En la base de datos, únicamente se almacena el nombre del archivo o la ruta relativa, en campos asociados a las entidades ImagenHistorial o Medida. El acceso a estas imágenes se realiza mediante peticiones HTTP autenticadas para las capturas, y el backend las sirve como recursos estáticos (tipo `image/jpeg`), respetando los permisos del usuario.

2.6 Fase de pruebas

2.6.1 Pruebas de validación de código

Durante esta fase se realizaron pruebas unitarias para garantizar que los métodos críticos del sistema funcionaran correctamente y de forma aislada. Se utilizaron herramientas como JUnit 5 y Mockito para simular comportamientos y aislar dependencias.

2.6.2 Pruebas de implantación en distintos soportes

Durante la fase de pruebas finales, se validó la correcta ejecución de la aplicación *Guía de Pesca Interactiva Asturiana* tanto en entornos locales como en producción. Esto permitió verificar su comportamiento real en diferentes condiciones de despliegue.

1. Pruebas en entorno local (localhost)

La aplicación fue ejecutada inicialmente desde entorno de desarrollo utilizando localhost tanto para el backend como para el frontend:

- El frontend desarrollado con **Ionic** fue ejecutado en el navegador (ionic serve).
- El backend Spring Boot fue levantado con <https://localhost:7228>.
- Se verificaron funcionalidades como:
 - Acceso y autenticación mediante JWT.
 - Visualización de mapas y puntos geográficos.
 - Recuperación y carga de imágenes.
 - Consulta del clima a través de la API de OpenWeather.
- Todas las funcionalidades respondieron correctamente en este entorno.

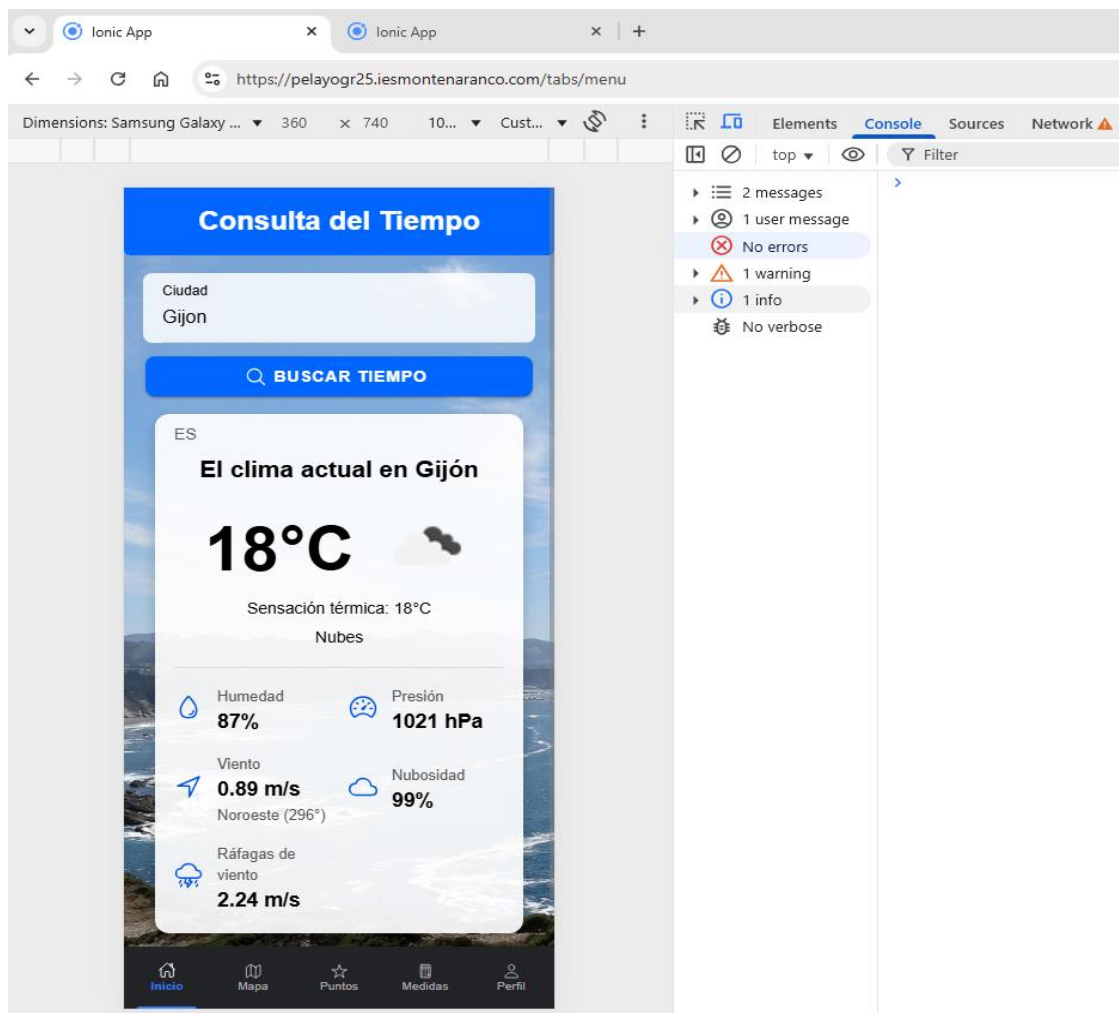
2. Pruebas en servidor remoto

Posteriormente, la aplicación fue desplegada en un servidor remoto accesible mediante HTTPS en la URL:

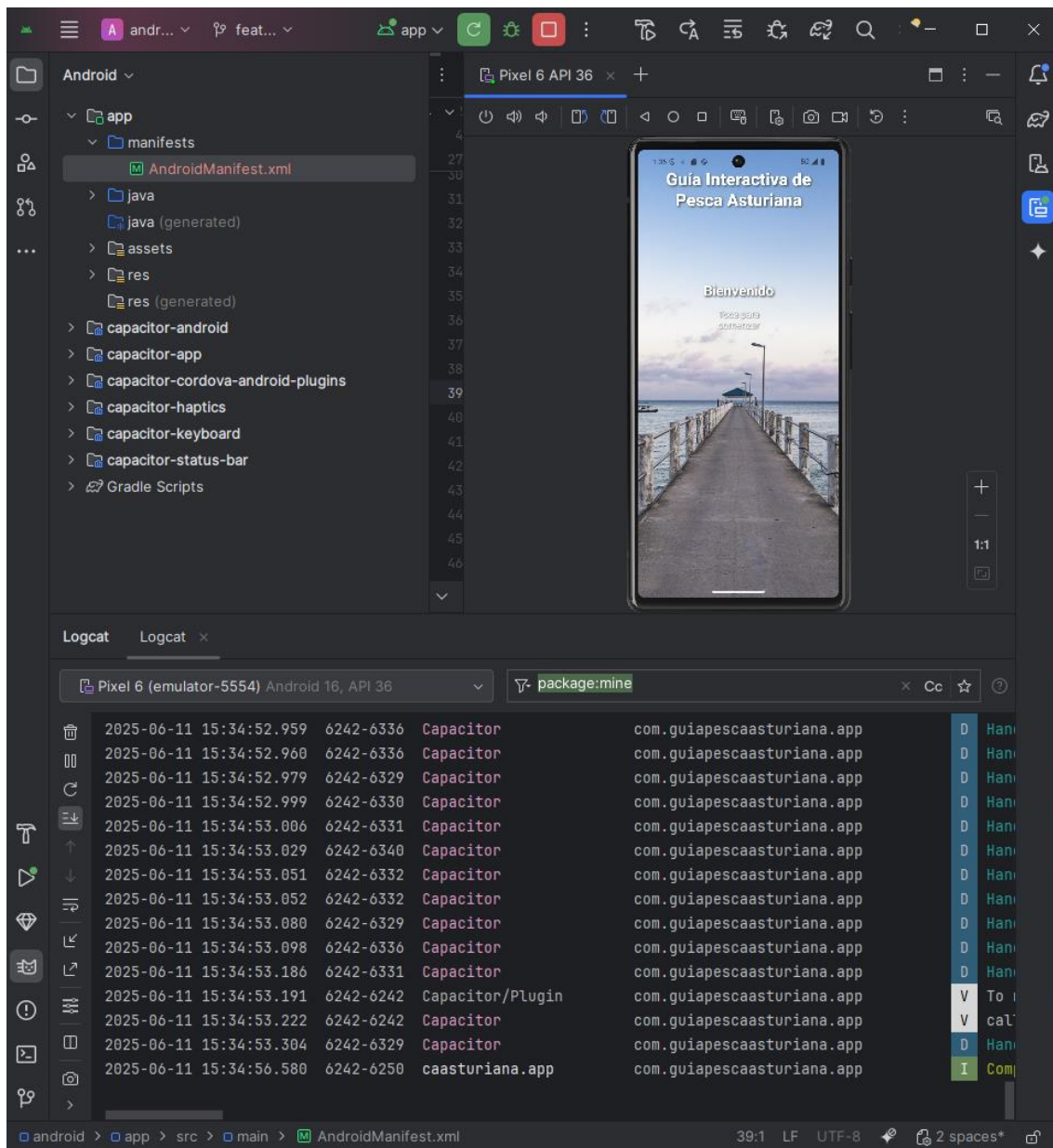
<https://pelayogr25.iesmontenaranco.com>

En este entorno se probaron las siguientes características, con resultados satisfactorios:

- Visualización desde navegadores modernos (Chrome, Firefox).
- Acceso al backend protegido con HTTPS.
- Correcto comportamiento del sistema de rutas.
- Carga dinámica de contenido (clima, puntos, medidas, imágenes).
- Comprobación del diseño **responsive** para dispositivos móviles.
- Pruebas desde el emulador de navegador (imagen adjunta: consulta del tiempo en Gijón).



3. Pruebas en emulador de Android Studio



Pruebas en emulador

4. Pruebas en APK

Se llevaron a cabo pruebas con diferentes móviles y versiones de APK, enviando a APK a amigos y familiares para sujetos de pruebas.

3. IMPLEMENTACIÓN

La aplicación se ha desarrollado utilizando **Ionic** en el frontend, aprovechando su compatibilidad con dispositivos móviles y su integración con tecnologías web modernas. Por su parte, el backend se ha construido con **Java** y **Spring Boot**, gestionando la autenticación mediante **JWT**, la seguridad mediante **H256** para contraseñas, el control de usuarios y roles (administrador/usuario), y la interacción con una base de datos relacional para el almacenamiento de puntos de pesca, registros históricos, imágenes, medidas por especie y otros datos relevantes. También se han integrado funcionalidades como la recuperación de contraseña por correo y el consumo de la API de OpenWeather para mostrar el clima actual.

El sistema está diseñado con una arquitectura segura, escalable y centrada en la experiencia del usuario, con un panel de gestión exclusivo para administradores, y funcionalidades visuales que incluyen mapas interactivos con **Leaflet**, formularios dinámicos, imágenes en galería y navegación intuitiva.

Para documentar la API se usó javadoc, podemos encontrar la documentación en el archivo `index.htm` dentro de `./apidocs/index.html` y para la documentación del Frontend se usó Compodoc, que se puede encontrar el `index.html` en la carpeta `./documentation/index.html`

4. PUESTA EN MARCHA. DESPLIEGUE

Para la puesta en marcha y despliegue hemos usado las siguientes herramientas :

- OpenVpn: <https://openvpn.net/>
- Termius : <https://termius.com/index.html>

Utilizamos **OpenVPN** para conectarnos por red al servidor y **Terminus** como interfaz para los contenedores de la aplicación API del backend Java/SpringBoot, para la aplicación del front Ionic/Angular y para la aplicación de la base de datos MySQL.

Necesitaremos crear y usar un **DockerFile** para el **Front** y **Backend**, un **Nginx** para la aplicación del **Fronted**, y **docker-compose.yml** para el contenedor de la base de datos.

```
FROM nginx:alpine

# Copiar la configuración personalizada de nginx
COPY nginx.conf /etc/nginx/conf.d/default.conf

# Copiar los archivos compilados localmente
COPY www/ /usr/share/nginx/html/

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```

DockerFile para el Front

```
server {
    listen 80;
    server_name localhost;
    root /usr/share/nginx/html;
    index index.html;

    # Configuración para Single Page Application
    location / {
        try_files $uri $uri/ /index.html;
    }

    # Configuraciones de cache para archivos estáticos
    location ~* \.(js|css|png|jpg|jpeg|gif|ico|svg)$ {
        expires max;
        add_header Cache-Control "public, max-age=31536000";
    }
}
```

Configuración Nginx para la aplicación Front

```
# Usar la imagen oficial de OpenJDK 21 como base
FROM eclipse-temurin:21-jdk

# Establecer el directorio de trabajo dentro del contenedor
WORKDIR /app

# Copiar el archivo JAR de la aplicación al contenedor
COPY target/guiadepesca-backend-o.o.1-SNAPSHOT.jar app.jar

# Exponer el puerto en el que corre la aplicación
EXPOSE 8080

# Comando para ejecutar la aplicación
ENTRYPOINT ["java", "-jar", "app.jar"]
```

DockerFile básico para el Backend

```
1  services:
2
3      # Frontend (Ionic/Angular)
4      frontend:
5          build:
6              context: ./frontend
7              dockerfile: Dockerfile
8          container_name: GuiaPesca-frontend
9          restart: always
10         ports:
11             - "80:80"
12         networks:
13             - GuiaPesca-network
14
15     mysql:
16         image: mysql:8
17         container_name: GuiaPesca-mysql
18         restart: always
19         environment:
20             MYSQL_ROOT_PASSWORD:
21             MYSQL_DATABASE: GuiaPesca
22             MYSQL_USER: pelayo
23             MYSQL_PASSWORD:
24         ports:
25             - "3306:3306"
26         volumes:
27             - ./initdb:/docker-entrypoint-initdb.d/
28             - mysql_data:/var/lib/mysql
29         networks:
30             - GuiaPesca-network
31
```

```
32     # Backend (Spring Boot)
33     backend:
34         build:
35             context: ./backend
36             dockerfile: Dockerfile
37         container_name: GuiaPesca-backend
38         restart: always
39         depends_on:
40             - mysql
41         environment:
42             SPRING_DATASOURCE_URL: jdbc:mysql://mysql:3306/GuiaPesca?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC
43             SPRING_DATASOURCE_USERNAME: pelayo
44             SPRING_DATASOURCE_PASSWORD:
45         ports:
46             - "8080:8080"
47         networks:
48             - GuiaPesca-network
49         volumes:
50             - ./images/capturas:/app/images/capturas
51             - ./images/medidas:/app/images/medidas
52
53     volumes:
54         mysql_data:
55     networks:
56         GuiaPesca-network:

```

*Configuración base **docker-compose.yml***

Para poder subir los archivos de nuestra aplicación Fronted usaremos el siguiente comando :

```
ionic build --prod
```

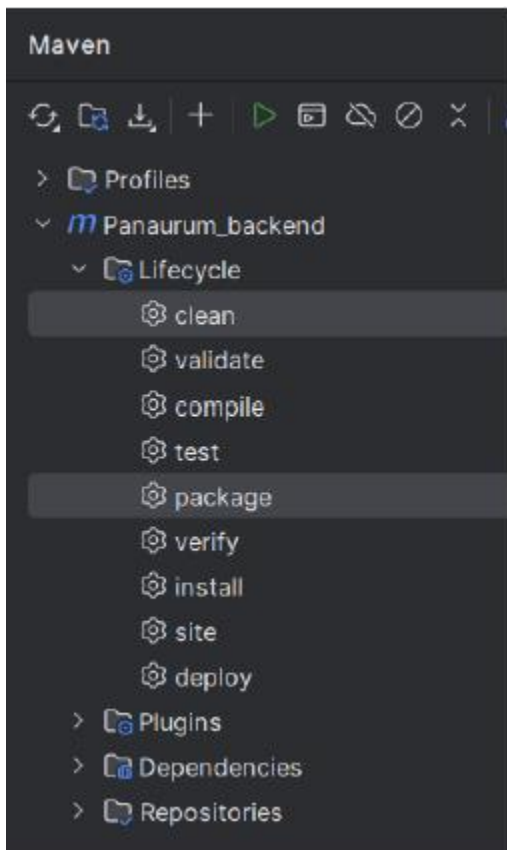
Una vez obtengamos los archivos generados en la carpeta `./www` lo subiremos al contenedor del terminus,

Para poder subir los archivos de nuestra aplicación Backend necesitaremos configurar el **application.properties**

```
7  # Configuraci3n JPA
8  #spring.jpa.hibernate.ddl-auto=update
9  spring.jpa.show-sql=true
10 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
11
12 # Configuraci3n JWT
13 jwt.secret=
14 jwt.expiration=3600000
15
16 # para correos
17 spring.mail.host=smtp.gmail.com
18 spring.mail.port=587
19 spring.mail.username=soportepescaasturiana@gmail.com
20 #Password segura del correo
21 spring.mail.password=
22 spring.mail.properties.mail.smtp.auth=true
23 spring.mail.properties.mail.smtp.starttls.enable=true
24
25 #OpenWeather
26 openweather.apikey=
27
28 #despliegue
29 #spring.datasource.url=jdbc:mysql://mysql:3306/GuiaPesca?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC
30 spring.datasource.url=jdbc:mysql://          :3306/GuiaPesca?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC
31 spring.datasource.username=pelayo
32 spring.datasource.password=
33 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
34 spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
35 spring.jpa.hibernate.ddl-auto=update
36
37 server.error.include-message=always
38 server.error.include-binding-errors=always
39 server.error.include-stacktrace=always
40
41 #para a1adir mas capacidad para subir imagenes
42 spring.servlet.multipart.max-file-size=10MB
43 spring.servlet.multipart.max-request-size=10MB
44 spring.web.resources.static-locations=file:./images/
```

Configuraci3n application.properties

Para generar los archivos para el contenedor del Backend usaremos :



Clean Para limpieza de caché



Package para generar el JAR

Una vez generado el JAR, lo subiremos al contenedor del Backend del termius.

5. CONCLUSION

5.1 Valoración personal del trabajo realizado.

La Guía de Pesca Interactiva Asturiana surge de la combinación entre mi experiencia personal como aficionado a la pesca y mi formación como desarrollador. Esto me ha permitido identificar con claridad las necesidades reales de los usuarios y diseñar una solución tecnológica adaptada al contexto local.

El desarrollo completo de esta aplicación me ha permitido consolidar conocimientos tanto en el diseño de arquitecturas frontend-backend como en el despliegue en entornos reales. En conjunto, ha sido una experiencia útil y aplicable profesionalmente.

5.2 Posibles ampliaciones

Mostar más condiciones climáticas, como el estado de las olas(metros, temperatura) condiciones marítimas(horas de la plena y bajamar) y fases lunares.

Posibilidad de subir vídeos al *Historial* , una recopilación de el momento en el cual se saca la captura del agua.

Agrupación de *Historiales* en temporadas, recoger los meses de marzo-junio como “*Temporada Sepias 2025*” para una mejor organización.

Personalización mayor al crear un historial, como poder marcar la fase lunar, condiciones atmosféricas y que la aplicación haga una media,

Página social donde los usuarios con nombre personalizable y/o anónimo pueden presumir de las imágenes de sus capturas y/o reaccionar a otras capturas de otros usuarios.

Mayor información legal para la pesca y captura de algunas especies que tienen medidas propias y diferentes por comunidad autónoma.

6. Bibliografía Y WEBGRAFÍA

6.1 Bibliografía

OpenWeather. *One Call API 3.0 Documentation*.

Disponible en: <https://openweathermap.org/api>

API empleada para consultar el estado meteorológico en la aplicación.

6.2 WEBGRAFÍA

Recursos gráficos utilizados en el frontend

Estas imágenes han sido incorporadas en el diseño visual de la interfaz, ya sea como fondos, cabeceras o elementos decorativos:

- Pixabay. *Mar, embarcadero, costa, muelle*. Disponible en: <https://pixabay.com/es/photos/mar-embarcadero-costa-muelle-cielo-3023812/>
- Pixabay. *Paisaje, puerto de Cudillero (Asturias)*. Disponible en: <https://pixabay.com/es/photos/paisaje-puerto-de-cudillero-asturias-1682374/>
- Pixabay. *Mar en Asturias*. Disponible en: <https://pixabay.com/es/photos/mar-asturias-nubes-costa-1366260/>
- Pixabay. *Playa en Asturias*. Disponible en: <https://pixabay.com/es/photos/playa-asturias-espaa-mar-cielo-4945630/>
- Pixabay. *Puerto de Cudillero*. Disponible en: <https://pixabay.com/es/photos/puerto-barcos-embarcaci%C3%B3n-cudillero-7144712/>
- Pixabay. *Panel de control*. Disponible en: <https://pixabay.com/es/photos/control-s-adentro-habitaci%C3%B3n-1853330/>
- Pixabay. *Barco de pesca*. Disponible en: <https://pixabay.com/es/photos/barco-de-pesca-pescador-mar-pesca-5736839/>
- Pixabay. *Anzuelo de pesca*. Disponible en: <https://pixabay.com/es/photos/gancho-pesca-anzuelo-de-pesca-7418284/>

Imágenes utilizadas con fines informativos (tallas mínimas por especie)

Estas fuentes han sido utilizadas para obtener o ilustrar las tallas mínimas de especies en Asturias:

- Gobierno del Principado de Asturias. *Peces autorizados en pesca recreativa*. Disponible en: <https://pesca.asturias.es/peces>
- Gobierno del Principado de Asturias. *Crustáceos autorizados*. Disponible en: <https://pesca.asturias.es/crust%C3%A1ceos>
- Gobierno del Principado de Asturias. *Moluscos autorizados*. Disponible en: <https://pesca.asturias.es/moluscos>
- Gobierno del Principado de Asturias. *Equinodermos autorizados*. Disponible en: <https://pesca.asturias.es/equinodermos>

7. Anexos

7.1 Imágenes

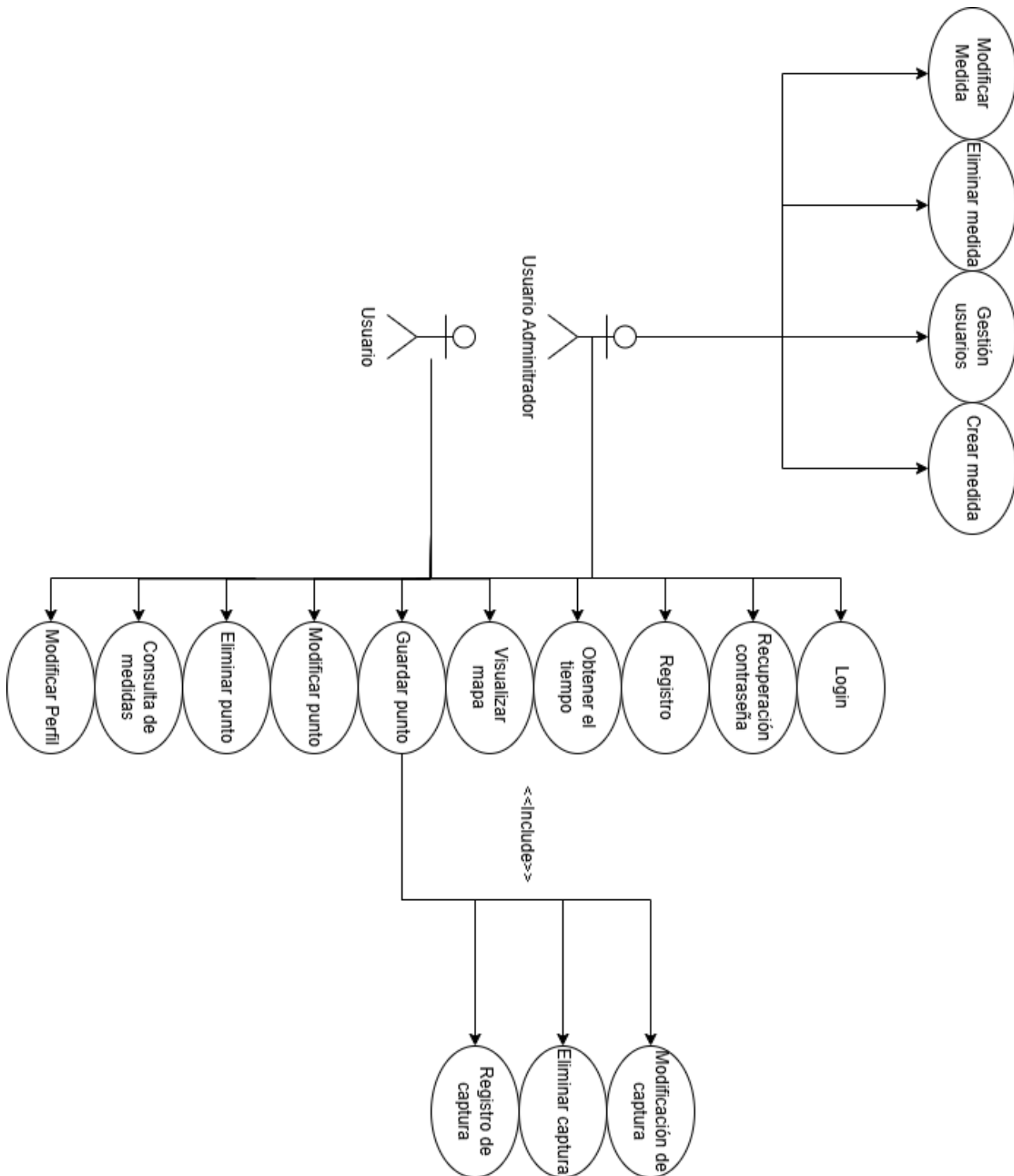
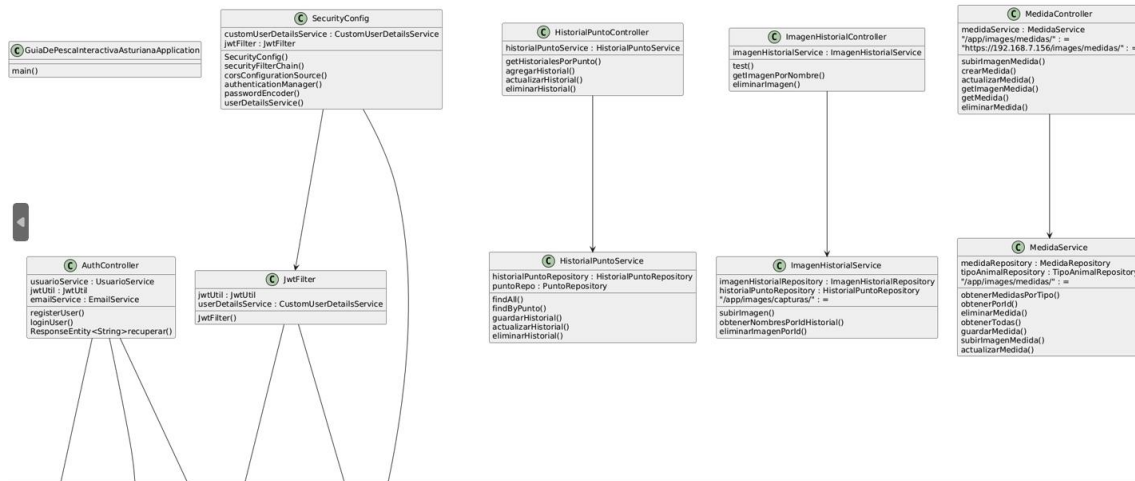


Diagrama de casos de uso



Diagrama de la base de datos MYSQL

Guía de pesca interactiva Asturiana



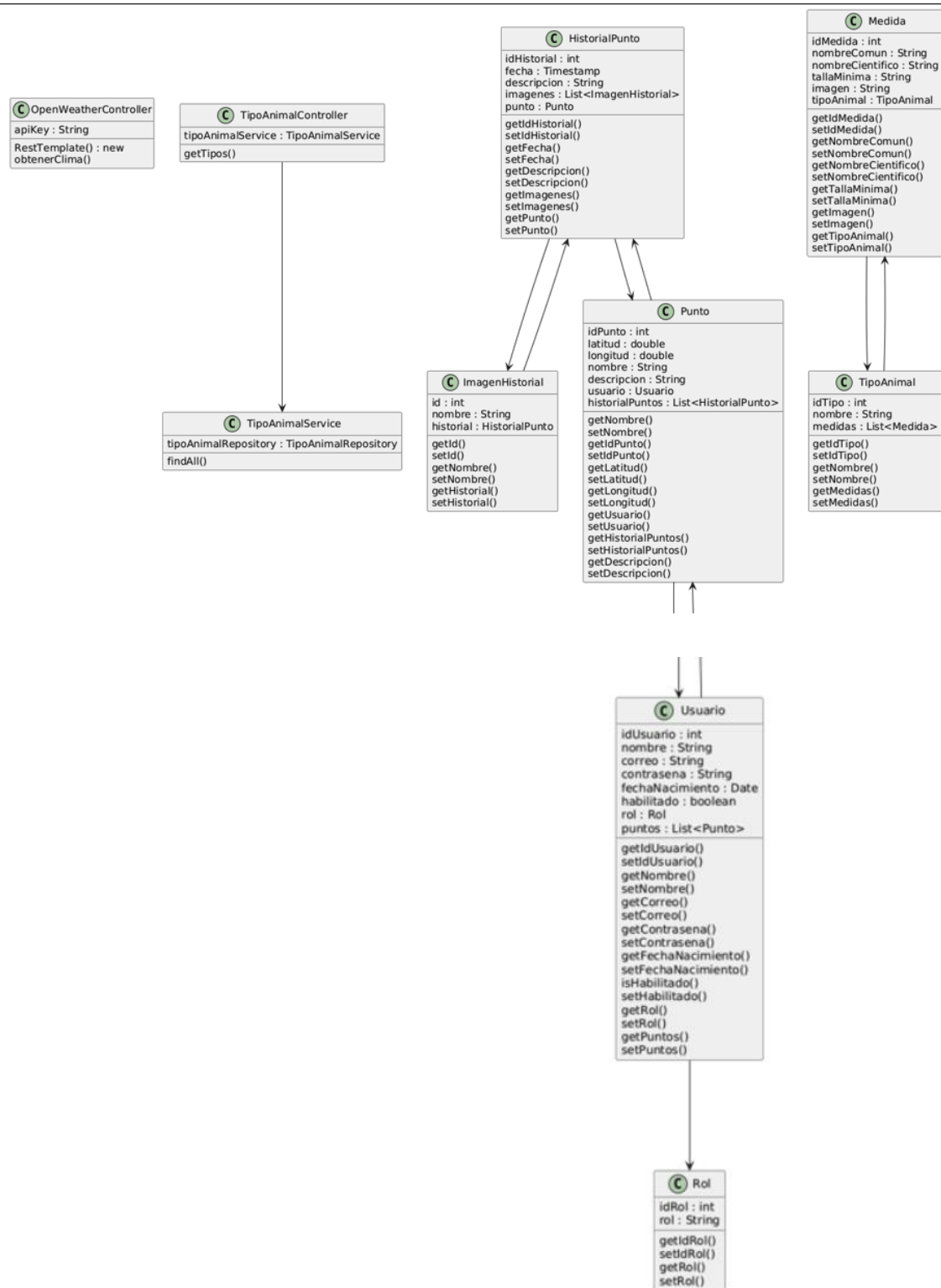


Diagrama de clases Backend



Diagrama de clases Fronted

7.2 Código completo del proyecto

Estructura del Front:

<https://github.com/DAM2A-PDAM-24-25/GuiaDePescaAsturiana-PelayoGonzalez.git>

Estructura del Back

<https://github.com/DAM2A-PDAM-24-25/GuiaDePescaAsturiana-PelayoGonzalez-backend.git>