



Software para refugios de animales

PeluNeko Adopciones

Acceso a Datos

2º CFGS a Distancia DAM

Eliu Manuel Viera Lorenzo

Contenidos

1.Introducción.....	Pág.4
2.Análisis.....	Pág.5
3.Diseño.....	Pág.11
4.Implementación.....	Pág.15
5.Conclusiones y trabajo futuro.....	Pág.16

1. Introducción

Desarrollo:

Lo que voy a desarrollar es una aplicación de escritorio para la gestión de animales abandonados, adopciones, voluntarios y nuevos dueños de un refugio de animales, para la gestión y control de los mismos animales a la hora de que reciban un hogar y saber a qué hogar van a estar.

Cada animal estará identificado con una ID principal conteniendo una foto guardada dentro de la base de datos

Funcionalidad:

La funcionalidad de esta sería que, dentro de los refugios, dado que son de personas civiles con la ayuda de instituciones tuvieran un control informatizado de los animales abandonados y los dueños que los adopten para tener un mayor control y no los abandonen de nuevo.

Motivación:

La motivación realmente no sabía que hacer, le dije a la IA (ChatGPT) dándome varias ideas para la misma me lanzo eso y dado que tengo dos animales en mi casa (Chihuahuas) y viendo todo lo que se encuentra en la calle de gatos y perros abandonados y las asociaciones que se lo llevan, me imagine que para mí sería un reto a la hora de hacerlo.

Dado que me gustaría implantar cosas que no había hecho anteriormente como las fotos de los animales a la hora del registro.

2. Análisis

La aplicación realizaría el registro de los animales que sean abandonados para tener un registro con su foto correspondiente para ver el estado del animal, que sería registrado por un voluntario del refugio de animales, para luego tener un registro de las adopciones y los nuevos propietarios de aquellos animales que sean adoptados, cada voluntario del refugio esta registrado y dispondrá de un login de inicio de sesión para poner usar la aplicación.

Entidades Principales

Voluntarios

Animales

NuevosDueños

Adopciones

Raza

LoginCentro

Los conceptos

Voluntarios: Personas que registran a los animales, nuevos dueños y las adopciones.

Animales: Mascotas rescatadas en espera de adopción.

NuevosDueños: Personas físicas que adoptan uno o más animales.

Adopciones: Registro que vincula animales adoptados con nuevos dueños y la fecha correspondiente.

Raza: Catálogo con las distintas razas de los animales.

LoginCentro: Registro del inicio de sesión de cada voluntario.

Especificación de Requerimientos

Requerimientos Funcionales (RF):

- **RF1.** Un voluntario con rol de administrador puede crear nuevos voluntarios y sus credenciales de acceso.
- **RF2.** Un voluntario registrado puede iniciar sesión en el sistema.
- **RF3.** Un voluntario puede registrar animales abandonados.
- **RF4.** Un voluntario puede registrar nuevos dueños.
- **RF5.** Un voluntario puede registrar adopciones entre animales y nuevos dueños.

Requerimientos No Funcionales (RNF)

- **RNF1.** Los datos se almacenarán en una base de datos segura.
- **RNF2.** Las contraseñas de los voluntarios deben estar cifradas de forma segura (por ejemplo, bcrypt o SHA-256 con sal).

Diagrama de Uso

Actores:

- Voluntario,
- NuevoDueño

Casos de uso:

- Iniciar sesión
- Registrar animal
- Registrar nuevo dueño
- Registrar adopción
- Crear nuevo voluntario (solo Administrador)

Diagrama:

Voluntario -> Crea un nuevo Voluntario (Solo Administrador)

Voluntario -> Iniciar Sesión

Voluntario -> Registra nuevo dueño

Voluntario -> Registra animal

NuevoDueño -> Registra adopción (Solo con DNI del objeto)

Detalle de un caso de uso:

Caso de uso: Registrar adopción

- **Actor:** Voluntario
- **Precondición:** El voluntario debe haber iniciado sesión. El animal y el nuevo dueño deben estar registrados.
- **Flujo principal:**
 1. El voluntario entraría en Adopciones
 2. El voluntario le daría al icono del +.
 3. El voluntario tendría que rellenar los datos del registro.
 4. El voluntario le daría al botón "Registro".
 5. La fecha de adopción se introduce automáticamente.
 6. El sistema guarda el registro de la adopción.
- **Postcondición:** La adopción queda registrada en la base de datos y todos los datos introducidos se pueden eliminar y modificar seleccionándolo en la misma tabla 2 clicks.

Caso de uso: Registrar Animal

- **Actor:** Voluntario
- **Precondición:** El voluntario debe haber iniciado sesión
- **Flujo principal:**
 1. El voluntario entraría en Animales
 2. El voluntario le daría al icono del +.
 3. El voluntario tendría que rellenar los datos del registro.
 4. El voluntario le daría al botón "Registro".
 5. El sistema guarda el registro del animal.
- **Postcondición:** La adopción queda registrada en la base de datos y todos los datos introducidos se pueden eliminar y modificar seleccionándolo en la misma tabla 2 clicks.

Diagrama de Clase:

Raza

-idRaza: int
-raza: String

Voluntario

-DniVoluntario: String
-NombreVoluntario: String
-Apellido1Voluntario: String
-Apellido2Voluntario: String
-DireccionVoluntario: String
-LocalidadVoluntario: String
-CodigoPostalVoluntario: String

Animales

- idAnimal: int
- TipoAnimal: String
- Peso: double
- Estado: String
- Imagen: Byte
- idRaza: Raza
- DniVoluntario: VoluntarioCentro

NuevosDuenos

- Dni: String
- Nombre: String
- Apellido1: String
- Apellido2: String
- Direccion: String
- Localidad: String
- CodigoPostal: String

loginCentro

- idLogin: int
- idVoluntario: VoluntarioCentro
- Password: String (Encriptada)
- Rol: String

Adopciones

-idAdopcion: int

-idAnimal: Animal

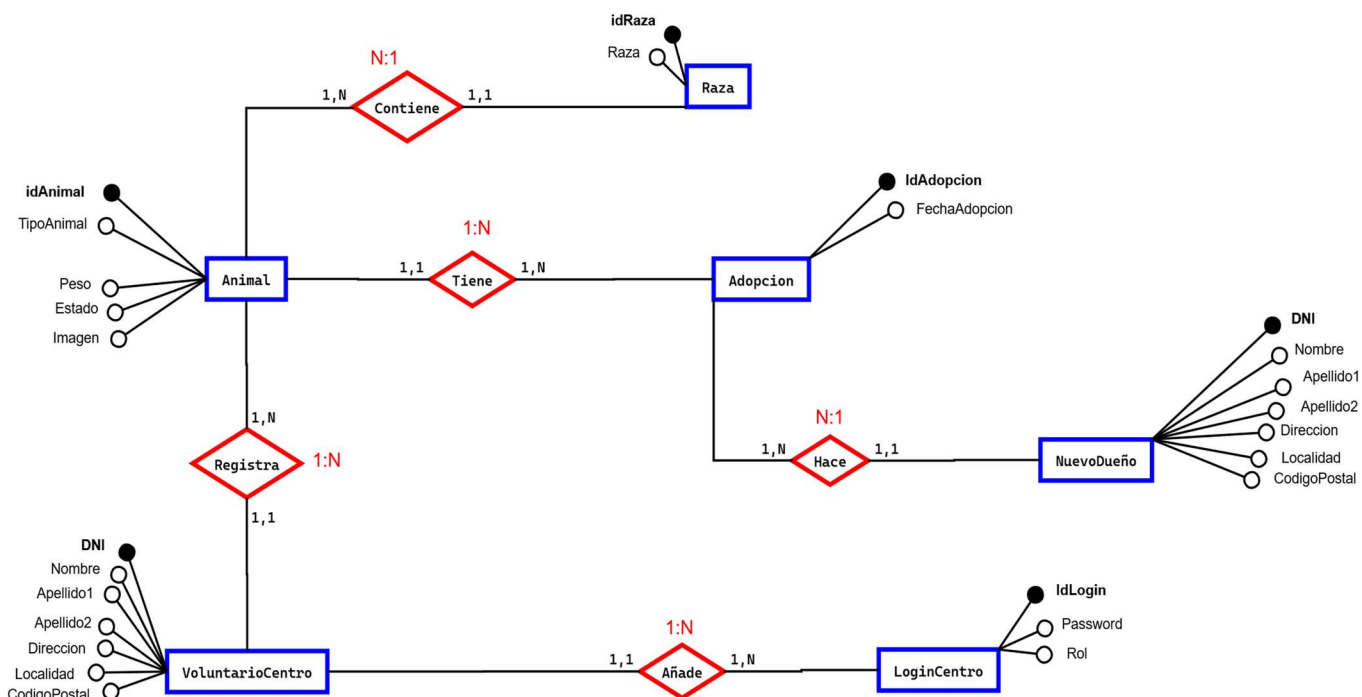
-dni: NuevosDuenos

-fechaAdopcion: Date

3. Diseño

Diseño de la base de datos

Modelo Entidad Relación



Modelo Relacional

Raza: **idRaza**, raza.

Animales: **idAnimal**, tipoAnimal, Peso, Estado, imagen, idRaza, DNIVoluntario*.

VoluntariosCentro: **DNIVoluntario**, NombreVoluntario, Apellido1Voluntario, Apellido2Voluntario, DireccionVoluntario, LocalidadVoluntario, CodigoPostalVoluntario.

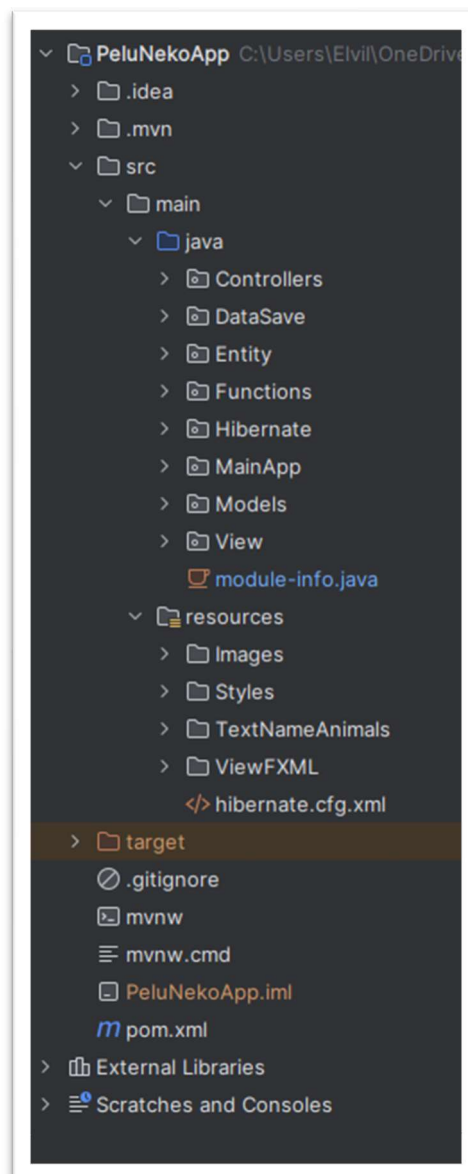
NuevosDuenos: **DNI**, Nombre, Apellido1, Apellido2, Direccion, Localidad, CodigoPostal.

LoginCentros: **idVoluntario***, password, Rol.

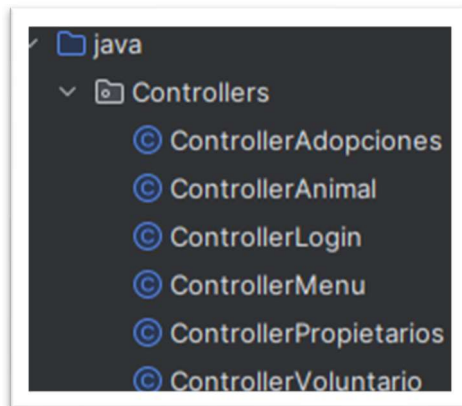
Adopciones: **idAdopcion**, FechaAdopcion, DNI*, idAnimal*.

Estructura de carpetas

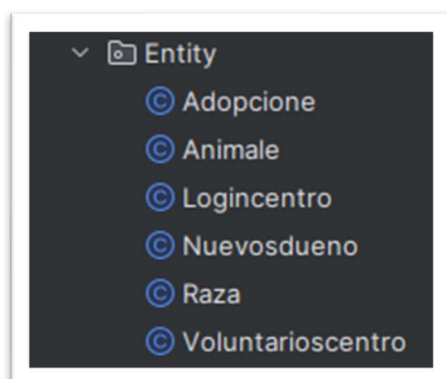
La estructura de carpetas del proyecto está realizada en MVC. Están divididas en dos partes la parte de código (JAVA) y la otra es la parte de diseño (resources). Aparte de las carpetas correspondientes del MVC nos encontramos **Funcions** conteniendo las funciones de las alertas y las aperturas de ventana de las FXML, DATASAVE (Para guardar los datos para los inicios de sesión siendo solamente como una clase más, pero publica). HIBERNATE (Conteniendo el script para las llamadas del ORM).



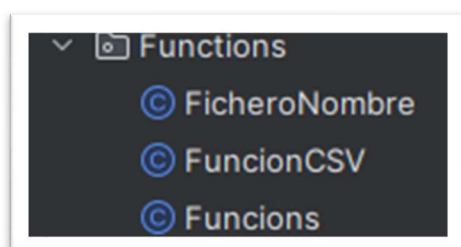
En la parte de java nos encontraremos los **CONTROLLERS** que serían todos los controladores que luego se llaman a los **VIEW** de cada **FXML**.



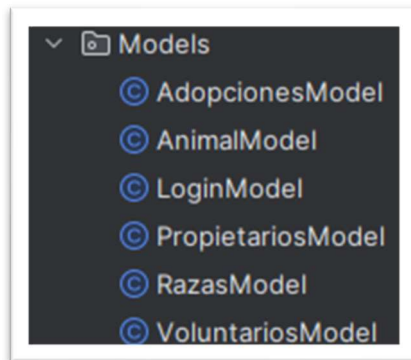
También nos encontraremos la carpeta de **ENTITY** que es donde tenemos todas las clases realizadas con el **ORM(HIBERNATE)** con sus datos correspondientes de la base de datos.



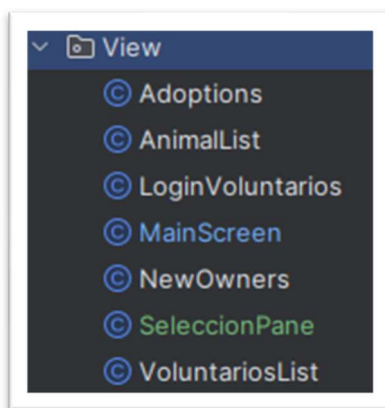
En la carpeta **FUNCTIONS** contendremos todas las funciones que estaremos utilizando en el proyecto para hacerlo más simplificado y ordenado



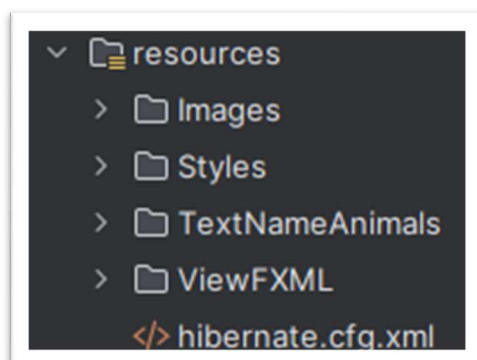
En la carpeta **MODELS** nos encontraremos todas las llamadas al ORM para luego llamarlas desde el controlador para tener mejor ordenador la estructura del proyecto.



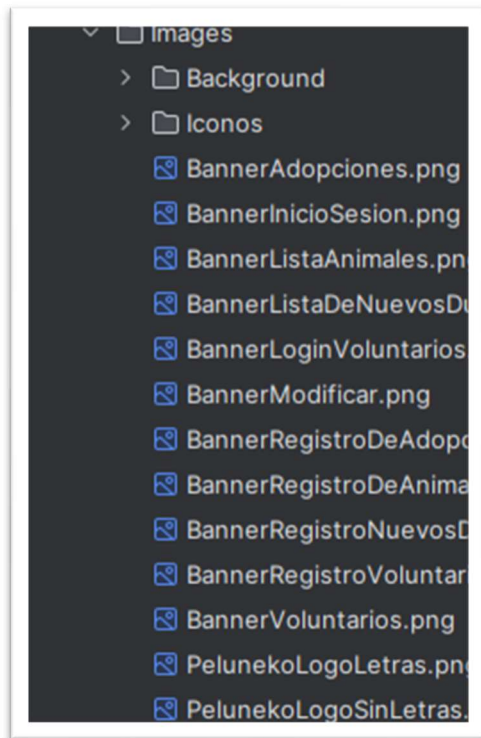
Por último, en la carpeta de **VIEW** nos encontraremos todas las vistas de los FXML para luego hacer las llamadas correspondientes a los controladores



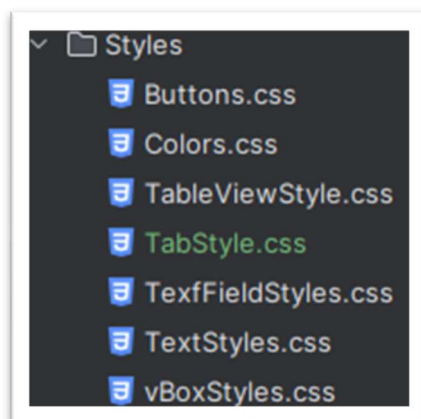
En el apartado de resources donde nos encontraremos toda la parte grafica del proyecto que se dividiría en:



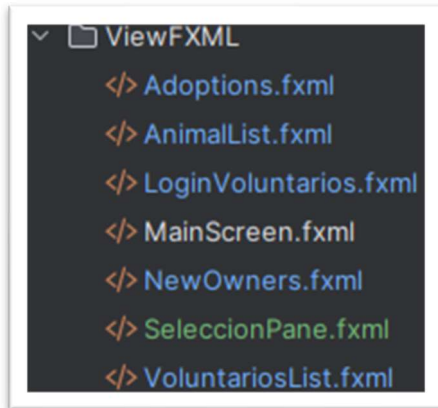
En la carpeta de IMAGES tendríamos todas las imágenes que tenemos en los proyectos divididos en Iconos, banners y logo.



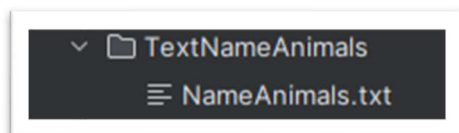
En la carpeta de STYLES, tendríamos todos los estilos CSS realizados para los FXML de JAVA FX.



En la carpeta de **VIEWFXML** es donde tendríamos todas las ventanas que estamos llamando al proyecto con su respectivo estilo llamado desde la carpeta anterior.

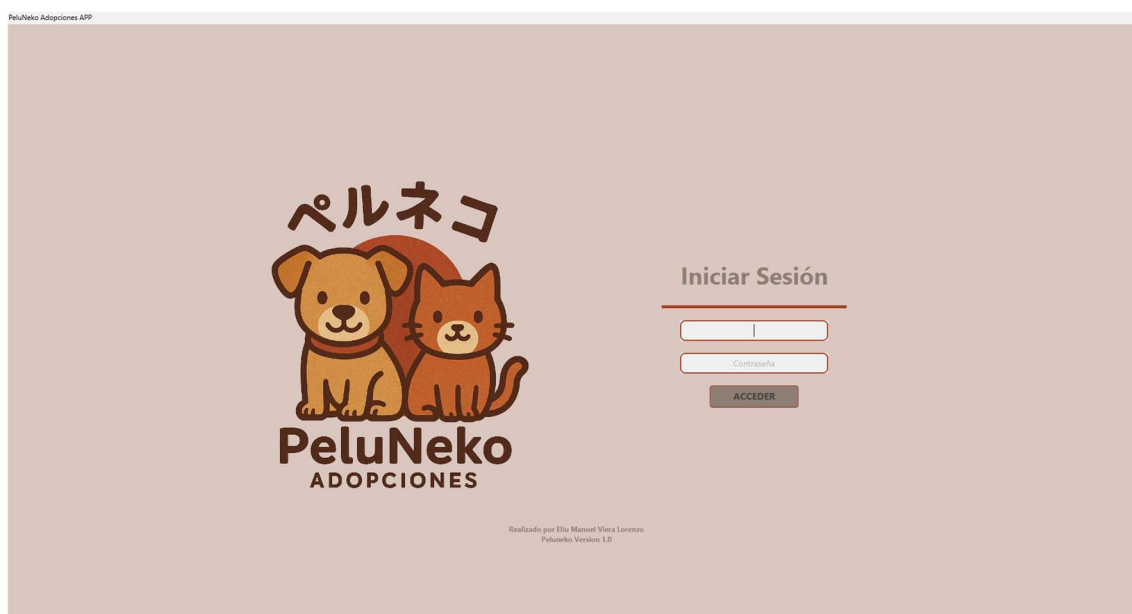


En esta carpeta contendremos los nombres de los animales que vamos generando de forma aleatoria.

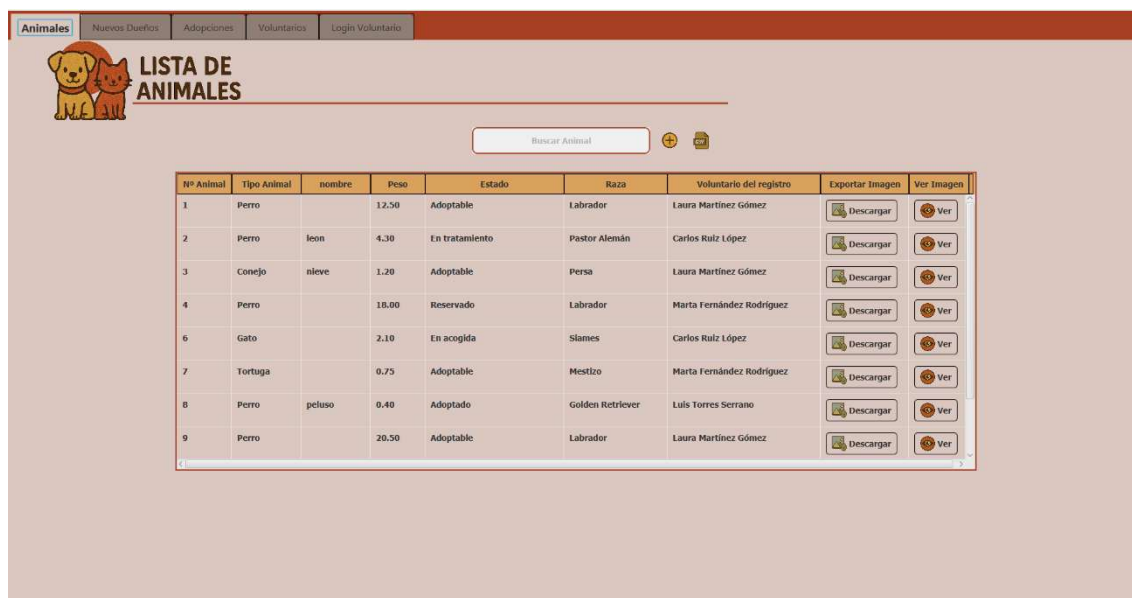


Diseño de la interfaz de usuario

Inicio:



ListaAnimales:



Nota: Dentro de la ventana de Lista de animales tendremos en la parte derecha para descargar y ver las imágenes que se añadan para registros el animal para ver el estado en que se encuentra.

[illegible][illegible]

Acceso a Datos. 2º CFGS a Distancia DAM
19

4. Implementación

Las Tecnologías usadas para el proyecto son:

Java 21:

Por la facilidad a la hora de usarlo con JavaFX dado que es el único lenguaje que lo hace y que ya lo conozco.

MySQL:

Mi elección es por qué siendo un modelo relacional a la hora de insertar los datos son más seguros a la hora de hacer las relaciones entre ellas para las adopciones y los nuevos propietarios.

JavaFX:

Mi elección es dado que es único framework grafica para java para realizar dicho proyecto, aunque lo hubiera elegido igualmente, por su facilidad a la hora de realizar las interfaces graficas por la facilidad del CSS para los estilos y la libertad de movimiento de las inserciones de los TextField, Botones, tablas, etc...

Inteligencia Artificial:

Para la creación de todas las imágenes implementadas en el proyecto.

PaintNET:

Por la facilidad a la hora quitar el fondo a las imágenes realizadas por la IA con la varita mágica

Paint Microsoft:

Para la modificación del tamaño de los banners e iconos

5. Conclusiones y Trabajo Futuro

Mi conclusión personal en este proyecto es, que es bastante complejo dado la implantología del **ORM** teniendo que modificar la base de datos porque tenía relaciones M:M teniéndolas que cambiar a 1:M y la parte de importar y exportar las imágenes de los animales a la base de datos se me complicó implementarlo y también la parte de los inicio de sesión de los voluntarios para entrar en la aplicación para que se pudiera ver los botones de voluntarios y login solo los usuarios con el rol de administrador.

Los posibles trabajos que se podrían implementar en el proyecto sería:

Realizar una base de datos en la nube para tener la facilidad de tenerlo en diferentes dispositivos sin necesidad de realizarlo en local.

Realizar una APP móvil para que los registros se puedan hacer sin problema sin necesidad de estar al frente de un ordenador para hacer el control de los animales.

Realizar a la hora de hacer un registro del animal realizar una impresión con su número de identificación y nombre en un chapita serigrafiada con un collar que se le pone al animal.

6. Bibliografía

Fuentes para los Banners y logos:

Imágenes de DALL-E 4 de OpenIA (ChatGPT)

Paint.net: <https://www.getpaint.net/> , Para quitarles los fondos a las imágenes

Paint: Microsoft, Para modificar los tamaños

Fuentes para los iconos:

Biblioteca de iconos: www.flaticon.es y posteriormente pasados por ChatGPT para hacer el mismo estilo que el realizado en los banners