

LostPaws

Motivación

Encontré a mi gato en la calle y esa experiencia me hizo ver cuántos animales perdidos o abandonados necesitan una segunda oportunidad. Esta app busca conectar a quienes los encuentran con quienes los buscan o desean adoptar, facilitando su rescate y promoviendo la adopción responsable. Mi objetivo es que más animales tengan la oportunidad de encontrar un hogar y una nueva vida.

Objetivo General

Desarrollar una aplicación móvil que facilite la localización y adopción de animales perdidos o abandonados, conectando a los usuarios de manera sencilla y eficiente.

Objetivos Específicos

1. **Facilitar el reencuentro de animales perdidos** con sus dueños a través de publicaciones y alertas en la app.
2. **Ofrecer un espacio para la adopción de animales** encontrados o no deseados, promoviendo la tenencia responsable.
3. **Registrar información de mascotas**, incluyendo vacunas, citas médicas y tratamientos.
4. **Permitir la comunicación directa entre usuarios** para agilizar adopciones y rescates.
5. **Integrar información sobre refugios y centros de adopción** para que los usuarios puedan encontrar opciones cercanas.
6. **Proporcionar información sobre veterinarios** para facilitar el acceso a ayuda sanitaria.
7. **Enviar notificaciones sobre situaciones críticas**, como refugios saturados o eventos de rescate.
8. **Incluir una sección de donaciones** a refugios y organizaciones, sin integración de pagos, solo informativa.

Metodología

Metodología Utilizada: KANBAN

Descripción:

Dado que el proyecto se realizará de manera individual, se ha optado por utilizar la metodología **Kanban**, la cual se adapta perfectamente a las necesidades de un trabajo autónomo. Kanban es un enfoque ágil que permite gestionar el flujo continuo de tareas mediante un tablero visual donde se organizan las actividades en diferentes etapas: **Pendiente, En Progreso y Hecho**. Esta metodología proporciona flexibilidad y permite adaptar las tareas según las necesidades del momento, sin la necesidad de seguir ciclos fijos de desarrollo como en otras metodologías ágiles.

Ventajas:

- **Flujo continuo:** Permite avanzar en las tareas de manera constante y organizada, sin la rigidez de ciclos de tiempo.
- **Facilidad de uso:** Es simple de implementar y entender, lo que resulta ideal para un único desarrollador. Además, se puede gestionar fácilmente mediante herramientas como Trello o Google Sheets.
- **Flexibilidad:** La capacidad de ajustar las prioridades o el enfoque del proyecto a medida que surjan nuevos desafíos o cambios.
- **Visualización del trabajo:** Facilita la **gestión visual** del progreso, asegurando que ninguna tarea se quede olvidada o se retrase innecesariamente.

Desventajas:

- Si no se gestiona correctamente, puede ser difícil **mantener el enfoque** en las tareas clave, ya que la estructura flexible no tiene un marco rígido de entregas o plazos fijos.
- Requiere disciplina para no dejar que el proyecto se des controle si hay demasiadas tareas en progreso al mismo tiempo.

Justificación:

Kanban es la opción más adecuada para este TFG, ya que permite un control claro del avance del proyecto y facilita la adaptación a cambios imprevistos. Como el trabajo se realizará de manera individual, se beneficiará de su simplicidad y flexibilidad, permitiendo avanzar de manera eficiente sin la necesidad de coordinación constante con un equipo.

Tecnologías y Herramientas Utilizadas

Para el desarrollo de esta aplicación, se han seleccionado herramientas y tecnologías que permiten crear un producto eficiente, moderno y escalable. A continuación, se detallan las principales elecciones y sus razones:

1. Lenguaje de Programación

- **Kotlin:** Se ha elegido por ser el lenguaje oficial para el desarrollo de aplicaciones Android. Ofrece una sintaxis concisa, seguridad frente a `NullPointerException` y compatibilidad total con Java, facilitando el desarrollo y mantenimiento del proyecto.

2. Entorno de Desarrollo

- **Android Studio:** Es el entorno de desarrollo integrado (IDE) oficial para Android. Proporciona herramientas avanzadas como emuladores, depuración en tiempo real y compatibilidad con Jetpack, lo que facilita la creación de interfaces modernas y funcionales.

3. Base de Datos y Almacenamiento

- **Firebase Realtime Database:** Se utilizará para almacenar la información de los animales, usuarios y publicaciones en tiempo real.
- **Google Drive API:** Se integrará para el almacenamiento de imágenes, permitiendo a los usuarios subir y acceder a fotos de animales de manera segura y eficiente. Google Drive ofrece escalabilidad y control sobre los archivos sin necesidad de mantener una infraestructura de almacenamiento propia.

4. Diseño y Edición Gráfica

- **Adobe Photoshop:** Se utilizará para la creación de logotipos, iconos e imágenes dentro de la aplicación, garantizando una interfaz visual atractiva y profesional.

5. Notificaciones y Servicios en la Nube

- **Firebase Cloud Messaging (FCM):** Permitirá el envío de notificaciones push a los usuarios, alertando sobre animales perdidos o situaciones críticas en su área.

6. Comunicación entre Usuarios

- **Firebase Authentication:** Se utilizará para gestionar el registro y autenticación de usuarios de forma segura mediante correo electrónico o cuentas de Google.

Justificación de la Elección

La combinación de estas herramientas permite desarrollar una aplicación robusta, escalable y fácil de mantener. Firebase elimina la necesidad de un backend complejo, mientras que Kotlin y Android Studio garantizan una experiencia de desarrollo fluida y eficiente. La integración con Google Drive API para el almacenamiento de imágenes ofrece una solución segura y confiable sin necesidad de gestionar un servidor propio.

Planning - Diagrama de Gantt

Está en un documento aparte → [TFG - Planning](#)

Análisis

Requisitos Funcionales

1. **Registro de Usuarios:** Los usuarios pueden registrarse con su correo o Google y acceder de manera segura.
2. **Subida y Visualización de Imágenes:** Los usuarios pueden subir y ver fotos de animales perdidos a través de Google Drive API.
3. **Publicación de Animales:** Los usuarios pueden crear publicaciones con detalles de animales perdidos o encontrados, incluyendo fotos y ubicación.
4. **Búsqueda y Filtrado:** Los usuarios pueden buscar y filtrar publicaciones por tipo de animal, ubicación, fecha, etc.
5. **Reencuentro de Animales Perdidos:** Los usuarios pueden contactar entre sí para devolver mascotas perdidas.
6. **Adopción:** Los usuarios pueden ofrecer animales para adopción y contactar a interesados.
7. **Gestión de Información Médica:** Los usuarios pueden registrar y consultar información médica de sus mascotas.
8. **Notificaciones Push:** Los usuarios recibirán notificaciones sobre animales cercanos o situaciones críticas.
9. **Comunicación entre Usuarios:** Los usuarios pueden comunicarse para facilitar adopciones y rescates.

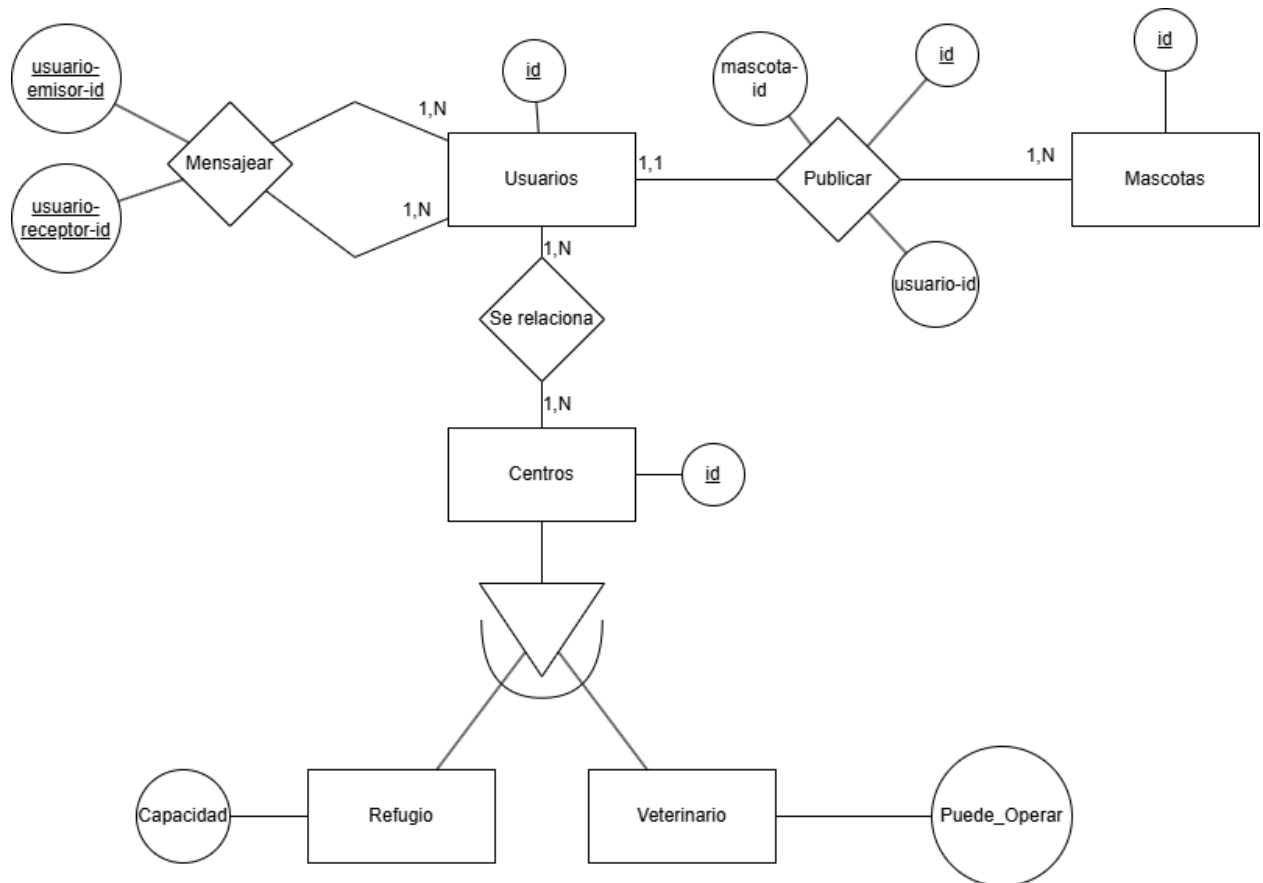
10. **Donaciones:** Los usuarios pueden ver opciones de refugios para donar, sin gestión de pagos.

Refugios y Veterinarios: Se incluirá información de refugios y veterinarios cercanos.

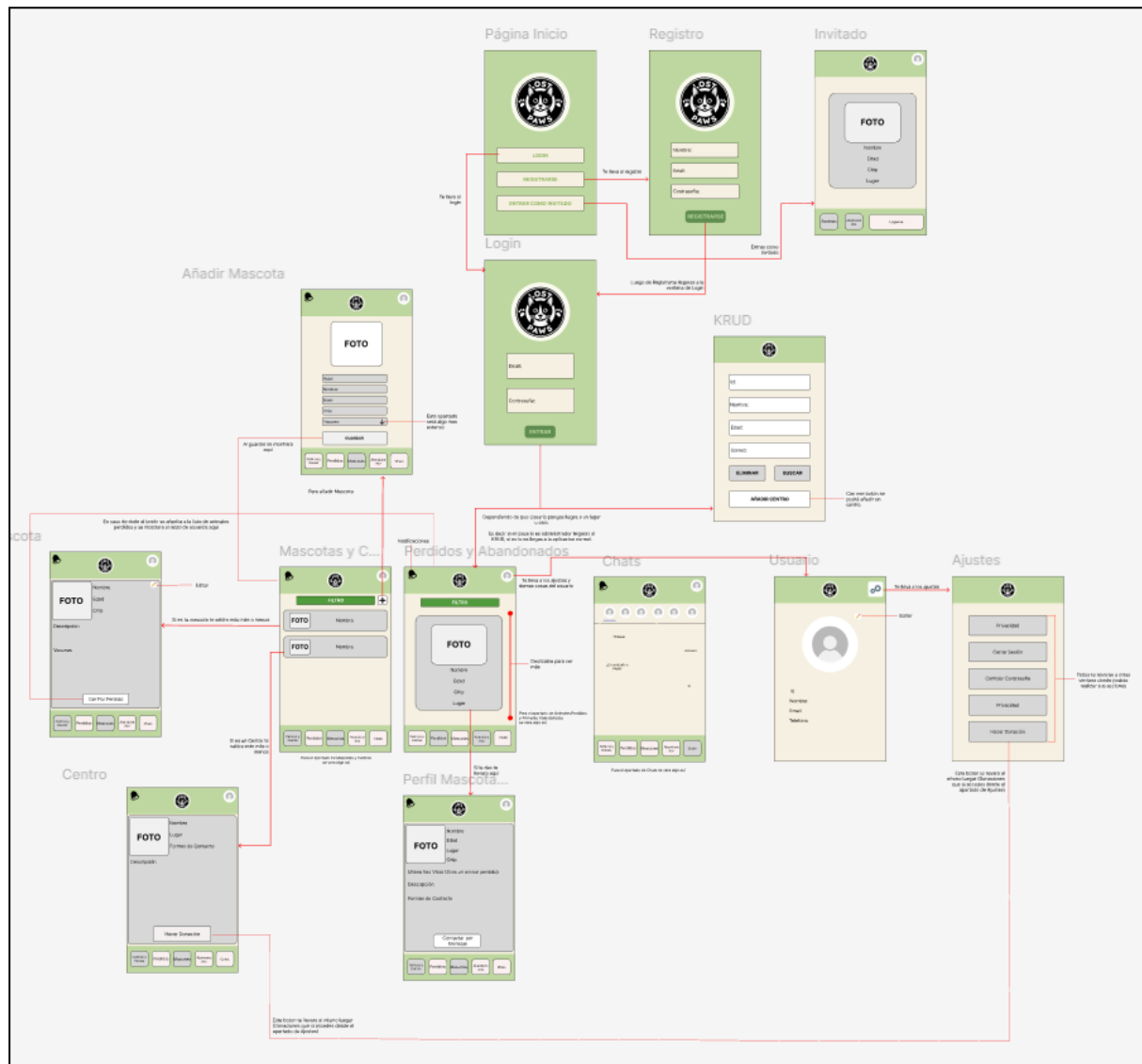
Requisitos No Funcionales

1. **Seguridad:** Uso de autenticación segura con Firebase y almacenamiento seguro de imágenes en Google Drive.
2. **Rendimiento:** Respuesta rápida y capacidad de manejar alto volumen de usuarios y datos.
3. **Escalabilidad:** Arquitectura escalable para futuras funcionalidades o más usuarios.
4. **Compatibilidad:** Compatible con dispositivos Android desde la versión 5.0.
5. **Usabilidad:** Interfaz intuitiva y fácil de usar, con diseño limpio.
6. **Disponibilidad:** La app debe estar disponible 24/7.
7. **Accesibilidad:** Accesible para personas con discapacidades.
8. **Backup y Recuperación:** Copias de seguridad automáticas para restaurar datos si ocurre un fallo.

Entidad Relación



Diseño



Para comprender mejor el **mockup**, es recomendable abrir el enlace y verlo en grande. Esto permite leer con claridad cada comentario y seguir las flechas, que indican a qué parte lleva cada elemento, facilitando así la comprensión del flujo y la interacción dentro del diseño.

Mock Up → [LostPaws](#)