

**Proyecto**

**“Pet Cholito”**

PTY4614-300D

San Joaquín

***Pet Cholito: Conectando corazones a través de la tecnología***

Cristhian Barreno

Nicolás Ruiz

Prof. Mariluz Rodiguez

30/11/2024

**ÍNDICE DE CONTENIDO**

**Abstract** [**3**](#_7j8xp3euar8b)

[**1. Información del proyecto. 3**](#_ctp81ey68jfe)

[1.1 Descripción del proyecto 3](#_48ht3hk3tfr4)

[1.2 Factibilidad el proyecto](#_mdulb4rdzphe) 4

[1.3 Modelo de Monetización](#_mdulb4rdzphe) 4

[1.4 Modelo de Negocio](#_mdulb4rdzphe) 4

[1.5 Validación Modelo de negocio](#_mdulb4rdzphe) 6

[**2. Perfil profesional.**](#_g0bypk6twcry) **7**

[2.1 Relación del proyecto con las competencias del perfil de egreso](#_sg329ra3da6j) 7

[2.2 Relación del proyecto con los intereses profesionales](#_ek0ifxvhv0sc) 7

[**3. Objetivos.**](#_4nz9lkqojdbd) **7**

[3.1 Objetivo general](#_kd3ooznjwuyz) 8

[3.2 Objetivo específico](#_72rik8bd7xmj) 8

**4**[**. Metodología de trabajo Scrum.**](#_4nz9lkqojdbd) **8**

4[.1 Justificación del uso de Scrum](#_kd3ooznjwuyz) 9

4[.2 Implementación de Scrum en el proyecto](#_72rik8bd7xmj) 9

4[.3 Planificación de Sprints](#_72rik8bd7xmj) 10

4[.4 Roles Y Responsabilidades](#_72rik8bd7xmj) 11

4[.5 Gestión de Riesgos y Mitigación](#_72rik8bd7xmj) 11

**5**[**. Competencias.**](#_4nz9lkqojdbd) **12**

5[.1 Competencias indirectas](#_kd3ooznjwuyz) 12

5.2 [Competencias directas](#_72rik8bd7xmj) 12

**6**[**. Evidencias.**](#_4nz9lkqojdbd) **13**

**7**[**. Stack Tecnológico.**](#_4nz9lkqojdbd) **14**

**8**[**. Diagrama Arquitectura.**](#_4nz9lkqojdbd) **15**

**9**[**. Plan de Trabajo.**](#_4nz9lkqojdbd) **15**

**10**[**. Historias de usuario.**](#_4nz9lkqojdbd) **18**

**11**[**. Base de datos.**](#_4nz9lkqojdbd) **18**

**12**[**. Carta Gantt.**](#_4nz9lkqojdbd) **23**

**13**[**. Plan de pruebas.**](#_4nz9lkqojdbd) **23**

13[.1 Tipos de pruebas](#_kd3ooznjwuyz) 12

13[.1 KPIS Pruebas](#_kd3ooznjwuyz) 12

**14**[**. Casos de pruebas.**](#_4nz9lkqojdbd) **24**

**15**[**. Seguridad.**](#_4nz9lkqojdbd) **26**

**16**[**. Flujo de caja Pet Cholito.**](#_4nz9lkqojdbd) **28**

**17**[**. Proyección de Escalabilidad de la Solución.**](#_4nz9lkqojdbd) **28**

**18**[**. Conclusión Individual.**](#_4nz9lkqojdbd) **29**

[Nicolás Ruiz](#_kd3ooznjwuyz) 29

Cristhian Barreno29

**19. Reflexión** **30**

# Abstract

**Español**

El proyecto "Pet Cholito" tiene como objetivo desarrollar una aplicación móvil que ayude en la búsqueda de mascotas perdidas, facilite el proceso de adopción y cree una comunidad amante de los animales. La aplicación permite a los usuarios registrar a sus mascotas, reportar pérdidas, y encontrar animales disponibles para adopción a través de un sistema de alertas y notificaciones. Además, incluye una red social donde los usuarios pueden compartir experiencias y recibir apoyo en temas relacionados con el bienestar animal y la visualización de distintos servicios animales como tiendas y veterinarias. Este proyecto no solo resuelve una problemática relevante en Chile sino que a nivel global, convirtiéndose en un proyecto altamente escalable.

**English**

The "Pet Cholito" project aims to develop a mobile application that helps locate lost pets, facilitates the adoption process, and creates a community of animal lovers. The app allows users to register their pets, report missing animals, and find pets available for adoption through an alert and notification system. Additionally, it features a social network where users can share experiences and receive support on topics related to animal welfare, as well as view various pet services like stores and veterinary clinics. This project addresses a relevant issue not only in Chile but also globally, making it a highly scalable initiative.

# 1. Información del proyecto.

Este proyecto integra diversas áreas de la informática, incluyendo desarrollo de aplicaciones móviles, gestión de bases de datos y redes sociales, demostrando cómo la tecnología puede aplicarse para abordar problemas sociales urgentes. A continuación, se explorarán en detalle los aspectos técnicos, funcionales y sociales de "Pet Cholito", así como su relevancia en el contexto chileno actual.

## **1.1 Descripción del proyecto**

“Pet Cholito" consiste en el desarrollo de una aplicación móvil que tiene como objetivo ayudar a las personas a encontrar mascotas perdidas, facilitar la adopción y crear una comunidad de amantes de los animales. La aplicación permitirá a los usuarios registrar a sus mascotas, reportar pérdidas y encontrar animales disponibles para adopción a través de alertas y notificaciones. Además, contará con una red social para subir contenido relacionado a los animales y recursos adicionales, como la visualización de tiendas de mascotas y veterinarias cercanas.

La relevancia del proyecto radica en el grave problema que enfrentan los animales sin hogar en Chile. Actualmente, se estima que hay aproximadamente 4 millones de perros y gatos sin hogar en el país. Además, solo en la comuna de Las Condes, se pierden alrededor de 600 perros cada mes. Estos números evidencian la necesidad urgente de una plataforma eficaz que conecte a los dueños de mascotas, personas interesadas en adoptar, y servicios relacionados con el cuidado animal. "Pet Cholito" contribuye tanto al bienestar animal como a la concientización social sobre la adopción y cuidado responsable.

## **1.2 Factibilidad el proyecto**

La evaluación de la factibilidad del desarrollo de "Pet Cholito" es crucial para asegurar que el proyecto pueda ser realizado de manera efectiva dentro de los límites de tiempo y recursos disponibles. Por lo tanto, varios factores han sido considerados para determinar si el proyecto es viable durante el semestre académico.

El desarrollo de Pet Cholito es factible considerando:

1. **Duración del semestre:** El proyecto se puede adaptar a la duración del semestre académico, enfocándose en desarrollar un MVP (Producto Mínimo Viable) con las funcionalidades esenciales.
2. **Horas asignadas a la asignatura:** Se puede ajustar el alcance del proyecto para que se adecue a las horas disponibles.
3. **Materiales requeridos:** Principalmente se necesitan computadoras y software de desarrollo, que están disponibles en la universidad o son de fácil acceso.
4. **Factores externos que facilitan su desarrollo:** Apoyo potencial de usuarios que apoyen el desarrollo de la aplicación.
5. **Factores externos que dificultan su desarrollo y soluciones:**
   * Limitaciones de tiempo: Priorizar funcionalidades clave para el MVP.
   * Falta de experiencia en aspectos específicos: Dedicar tiempo al aprendizaje autodidacta y buscar mentoría.
   * Desafíos en la adopción por usuarios: Desarrollar una estrategia de marketing enfocada en la comunidad local.
   * Costos de softwares: Herramientas como Firebase y Google Cloud tienen servicios con costos.

## **1.3 Modelo de Monetización**

El modelo de monetización de "Pet Cholito" está diseñado para generar ingresos a través de varias fuentes estratégicas, aprovechando tanto las características y funcionalidades de la aplicación como su expansión y crecimiento. A continuación se detallan las principales estrategias de monetización:

1. **Rango de Notificaciones Premium**: La aplicación ofrece un rango de notificaciones gratuito limitado a 5 kilómetros. Los usuarios que deseen ampliar este rango podrán optar por una suscripción premium que les permita aumentar la cobertura de las alertas a áreas más extensas.
2. **Perfiles de Mascotas Adicionales**: Los usuarios pueden crear un perfil gratuito para una mascota. Para aquellos que tienen múltiples mascotas, se ofrecerá la opción de pagar una tarifa adicional por perfil extra.
3. **Colaboraciones y Publicidad Local:** "Pet Cholito" establecerá asociaciones con veterinarias, tiendas de mascotas y otros negocios relacionados. Estas colaboraciones pueden incluir promociones dentro de la aplicación, descuentos especiales para usuarios, y anuncios patrocinados.
4. **Anuncios en la Aplicación:** La aplicación incorporará un sistema de anuncios, permitiendo a empresas relevantes publicar anuncios dirigidos a los usuarios. Estos anuncios podrán ser visualizados en varias secciones de la aplicación, como en la página de inicio, durante la navegación en la red social y en las alertas.
5. **Planes de Suscripción Premium:** Se ofrecerán varios planes de suscripción premium que incluirán beneficios adicionales, como la eliminación de anuncios y acceso a funcionalidades exclusivas.

## **1.4 Modelo de Negocio**

El modelo de negocio de Pet Cholito se basa en ofrecer una aplicación móvil que se enfoca en crear una comunidad de personas amantes de los perros y gatos.

* Segmentos de Clientes: Está dirigida a personas de 14 a 65 años interesadas en encontrar o adoptar mascotas.
* Relaciones con el Cliente: Ofrece soporte en la aplicación para resolver dudas y problemas.
* Canales: Usa redes sociales como Tik Tok, Instagram y tiendas de aplicaciones (App Store y Play Store) para llegar al público.
* Actividades y Recursos Clave: Las actividades principales incluyen la publicación de alertas y la promoción en redes. Los recursos clave son el equipo de desarrollo y marketing.
* Socios Clave: Veterinarias, empresas de alimentos y tiendas de accesorios para mascotas.
* Estructura de Costes: Incluye gastos en personal, servidores, bases de datos, dominio y publicidad.
* Fuentes de Ingresos: Genera ingresos mediante anuncios, patrocinios y suscripciones para establecimientos y usuarios.



## **1.5 Validación de Modelo de Negocio**

Para validar el modelo de negocio de Pet Cholito, se realizaron actividades que permitieron comprender mejor las necesidades y expectativas de los usuarios.

* Historias de Usuario: Se crearon y definieron 18 historias de usuario. Estas historias representan los diferentes requisitos funcionales y no funcionales que los usuarios desearían de la aplicación, cubriendo aspectos como creación de alertas, seguimiento de mascotas, gestión de perfiles, etc.
* Entrevistas a Personas: Se entrevistaron a 10 personas para obtener información directa sobre sus experiencias y expectativas en el ámbito de mascotas, especialmente en lo relacionado con la búsqueda, adopción, y atención de mascotas. Estas entrevistas ayudaron a refinar las características del producto y a ajustar la propuesta de valor.
* Encuesta: Se realizó una encuesta para recopilar datos cuantitativos sobre las preferencias y hábitos de los usuarios potenciales. Los resultados permitieron identificar oportunidades para asegurar que la aplicación sea útil para la comunidad de amantes de mascotas.

Para Terminar de validar el modelo de negocios, puedes revisar los procesos de hipótesis y ver las fotos de Google Forms en los siguientes enlaces:

1.- [Proceso hipotesis Pet Cholito](https://docs.google.com/document/d/1nuQ_hhKilHV_tG7tBOrNORUf0iaA9XLqOyHHcwl5Kp8/edit?tab=t.0)

2.-[Evidencia Google Forms.pdf](https://drive.google.com/file/d/1VVvERuEpFuYsf9REoQolzbxuYFJOU293/view?usp=drive_link)

# 2. Perfil Profesional

Este apartado aborda cómo el proyecto APT, en este caso "Pet Cholito", se relaciona con las competencias y habilidades que los estudiantes desarrollan a lo largo de su carrera, en función del perfil de egreso de la Ingeniería Informática. Se describe cómo el proyecto pone en práctica conocimientos adquiridos en diversas áreas, como el desarrollo de software, la gestión de proyectos, y las tecnologías emergentes como la ciberseguridad y la computación en la nube. Además, se explica la relevancia del proyecto para los intereses profesionales de los desarrolladores, alineando los objetivos del proyecto con su futuro en el campo laboral.

## **2.1 Relación del proyecto con las competencias del perfil de egreso**

El proyecto "Pet Cholito" está directamente vinculado con las competencias del perfil de egreso de Ingeniería Informática, que incluye habilidades clave como el diseño y desarrollo de software, la gestión de bases de datos, la arquitectura en la nube y la programación en lenguajes, como JavaScript. Además, el proyecto exige una implementación robusta de ciberseguridad para la protección de datos sensibles de los usuarios. La planificación y ejecución del proyecto también abarcan la gestión de proyectos, permitiendo una experiencia integral que combina conocimientos técnicos y habilidades de liderazgo. El desarrollo de esta aplicación móvil no solo refuerza las competencias técnicas de los egresados, sino que también ofrece una oportunidad para poner en práctica el emprendimiento tecnológico, siendo un proyecto altamente escalable y con impacto social.

## **2.2 Relación del proyecto con los intereses profesionales**

Nuestros intereses profesionales se enfocan en áreas claves para el mercado, como el desarrollo de software móvil, la ciberseguridad, y la gestión de proyectos tecnológicos. El proyecto "Pet Cholito" refleja estos intereses al proporcionar un entorno en el que podemos aplicar nuestras habilidades en el diseño y desarrollo de aplicaciones móviles a gran escala. Además, la aplicación requiere la implementación de medidas de ciberseguridad para garantizar la protección de datos, lo que fortalece nuestras competencias en este campo. El proyecto nos permitirá crecer profesionalmente al enfrentar desafíos reales en el desarrollo de soluciones tecnológicas con impacto social, contribuyendo no solo al bienestar animal, sino también a nuestra formación en gestión de proyectos y emprendimiento.

# 3. Objetivos

Para alcanzar el éxito en el desarrollo de "Pet Cholito," es fundamental establecer objetivos claros que guíen todas las fases del proyecto. Estos objetivos se dividen en un objetivo general y varios objetivos específicos que describen los pasos concretos necesarios para lograr el propósito principal del proyecto.

## **3.1 Objetivo general**

El objetivo del proyecto es desarrollar una aplicación móvil, competente según los estándares del mercado y que se pueda establecer como un líder en lo que es la ayuda a mascotas, promover la adopción responsable y fomentar una comunidad comprometida con el bienestar animal.

## **3.2 Objetivo específico**

Para alcanzar este objetivo general, se han definido varios objetivos específicos que orientan el desarrollo y la implementación de la aplicación:

1. **Implementar perfil de registro de mascotas:** Registrar a sus mascotas de manera detallada, para permitir afiliar mascotas a tu usuario.
2. **Reporte de pérdida y adopción:** Crear alertas en un mapa que permita a los usuarios reportar la pérdida de una mascota o la disponibilidad de una mascota para adopción, con opciones para agregar fotos, ubicaciones, descripciones detalladas, etc.
3. **Desarrollar un sistema de notificaciones:** Implementar un sistema de notificaciones que informe a los usuarios sobre reportes de mascotas perdidas o disponibles para adopción en su área, aproximadamente 10 KM.
4. **Crear una red social en la aplicación:** Desarrollar una sección dentro de la aplicación que permita a los usuarios compartir experiencias, consejos, y actualizaciones sobre sus mascotas, fomentando la participación activa de la aplicación.
5. **Autenticación con Google y Facebook:** Permitir a los usuarios autenticarse con google o facebook facilitará el uso de la aplicación.
6. **Visualizar servicios de mascotas:** Permitir a los usuarios ver cuáles servicios de mascotas como veterinarios, tiendas de alimentos, refugios de animales, etc. Se encuentran cerca de la ubicación del usuario y poder buscar en caso de preferencia.

# 

# 4. Metodología de trabajo Scrum

El enfoque metodológico propuesto para el desarrollo de "Pet Cholito" es Scrum, un marco ágil altamente eficaz para proyectos de software que permite una planificación flexible, adaptativa y orientada a resultados en entornos cambiantes. Scrum se basa en la iteración continua y la entrega de incrementos funcionales del producto a través de Sprints, lo cual es fundamental para un proyecto como "Pet Cholito", que abarca múltiples funcionalidades y requiere adaptabilidad tanto técnica como de gestión.

## **4.1 Justificación del uso de Scrum**

Scrum es una metodología que se adapta perfectamente a los requerimientos del desarrollo de una aplicación móvil como "Pet Cholito", debido a su naturaleza iterativa y su capacidad para gestionar cambios de manera eficiente. Este marco ágil permite desarrollar un producto funcional en fases cortas, lo cual es esencial considerando que el proyecto debe ajustarse a las limitaciones de tiempo del semestre académico. Además, permite enfocarse en la creación de un **Producto Mínimo Viable (MVP)** con las funcionalidades clave, como el registro de mascotas, notificaciones de animales perdidos, la red social y chat.

Scrum facilita también la priorización de tareas, con el objetivo de entregar el mayor valor posible al usuario final, especialmente cuando se enfrenta un calendario ajustado y recursos limitados. Al dividir el proyecto en Sprints, se asegura que el equipo se mantenga enfocado en metas específicas, alcanzables, y en constante mejora, adaptando el proyecto en función de los avances y las retroalimentaciones obtenidas a lo largo del tiempo.

## **4.2 Implementación de Scrum en el proyecto**

1. **Estructura del equipo de Scrum**

El equipo de desarrollo se dividirá en tres roles clave, pero aspectos del proyecto los miembros realizarán misma tareas en algunos casos:

* **Product Owner (PO):** Será responsable de maximizar el valor del producto y gestionar el **Product Backlog**, priorizando las tareas según su valor para el usuario y los objetivos del proyecto.
* **Scrum Master:** Facilitará el proceso Scrum, asegurando que las prácticas ágiles se sigan correctamente, y ayudará a remover cualquier obstáculo que interfiera con el progreso del equipo.
* **Development Team:** El equipo de desarrollo estará compuesto por los desarrolladores encargados de la programación de la aplicación, implementación de la base de datos, y el desarrollo de la infraestructura en la nube, incluyendo la integración de medidas de ciberseguridad y el diseño de la experiencia de usuario (UX/UI).

1. **Sprint Planning:**

Los Sprint tendrán una duración de entre 2 y 6 semanas, lo cual permite planificar y desarrollar funcionalidades del proyecto de manera incremental. En cada **Sprint Planning**, el equipo seleccionará las tareas del **Product Backlog** que se llevarán a cabo en ese Sprint, basadas en la prioridad establecida por el Product Owner. Las funcionalidades clave para los primeros Sprints incluirán:

* Alertas de mascotas perdidas y de adopción
* Vista de alertas
* Creación de perfiles para mascotas.
* Login, google auth y facebook auth.
* Subida de contenido para red social
* Chat
* Servicios
* Subscripciones

1. **Daily Scrum:**

Se realizará una breve reunión diaria para alinear el progreso del equipo, identificar posibles obstáculos y planificar las actividades del día. Estas reuniones asegurarán que el equipo esté sincronizado y que se mantenga el enfoque en las tareas más importantes.

1. **Sprint Review:**

Al final de cada Sprint, se llevará a cabo una Sprint Review donde el equipo presentará el incremento del producto a los interesados (Usuarios), mostrando lo que se ha completado y obteniendo retroalimentación que será incorporada en los siguientes Sprints. Este enfoque permitirá ajustar el producto de acuerdo a las necesidades reales de los usuarios y los objetivos del proyecto.

1. **Sprint Retrospective:**

Al final de cada Sprint, el equipo participará en una Sprint Retrospective, donde analizarán qué salió bien, qué se puede mejorar, y qué obstáculos se enfrentaron. Este proceso es clave para mejorar continuamente el flujo de trabajo, la colaboración y la productividad del equipo, lo que resulta en una mayor calidad del producto final.

1. **Backlog Refinement:**

El Product Backlog será continuamente refinado para asegurar que las tareas estén bien definidas, priorizadas y estimadas. Las funcionalidades más críticas, como la seguridad en la protección de datos, el sistema de alertas en tiempo real y la red social serán priorizadas en el backlog.

## **4.3 Planificación de Sprints**

A lo largo del semestre académico, se proyecta la ejecución de aproximadamente 5 Sprints, cada uno centrado en diferentes módulos o funcionalidades de la aplicación. Algunos de los hitos clave incluyen:

* **Sprint 1:** Desarrollo del MVP, con funcionalidades de crear alertas en el mapa, login, vistas de red social y mockups.
* **Sprint 2:** Desarrollo de documentación, google auth y creación de perfiles de mascotas.
* **Sprint 3:** Facebook, chat, subida de contenido y alertas en dispositivos.
* **Sprint 4:** Lógica de likes, comentarios, lógica para veterinarias y tiendas de mascotas.
* **Sprint 5:** Subscripciones, validación de seguridad, validación de lógicas.

## **4.4 Roles y Responsabilidades**

Esta sección detalla los roles específicos asignados dentro del equipo y sus responsabilidades principales, complementando la metodología Scrum descrita previamente.

| **Rol** | **Responsabilidades Principales** | **Asignado a** |
| --- | --- | --- |
| Product Owner (PO) | - Definir y priorizar los elementos del Product Backlog.  - Maximizar el valor del producto para los usuarios.  - Actuar como enlace entre el equipo de desarrollo y los stakeholders. | Nicolás Ruiz |
| Scrum Master | - Facilitar las ceremonias Scrum (Daily Scrum, Sprint Planning, Sprint Review, Sprint Retrospective).  - Asegurar que el equipo siga las prácticas ágiles.  - Eliminar impedimentos. | Cristhian Barreno |
| Equipo de Desarrollo | - Implementar las funcionalidades del Product Backlog.  - Diseñar y desarrollar la arquitectura del sistema.  - Realizar pruebas unitarias e integradas para asegurar calidad. | Nicolás Ruiz, Cristhian Barreno |

## **4.5 Gestión de Riesgos y Mitigación**

La gestión de riesgos es un aspecto fundamental en la ejecución del proyecto, ya que permite identificar, evaluar y mitigar los posibles desafíos que puedan afectar su desarrollo. A través de un enfoque proactivo, se han implementado estrategias y acciones para reducir el impacto de los riesgos identificados y garantizar el éxito del proyecto.

Para consultar la Gestión de Riesgos y Mitigación completo, accede al siguiente enlace:

[PlanGestiondeRiesgos.docx](https://docs.google.com/document/d/1OKI1i-P_lMu3kV0TEjGf06thq9iUsfmK/edit?usp=drive_link&ouid=117012098905063077400&rtpof=true&sd=true)

# 5. Competencias

Al analizar el panorama competitivo para "Pet Cholito", es fundamental identificar tanto las competencias directas como indirectas para entender mejor nuestro posicionamiento en el mercado y cómo podemos destacar frente a ellas. A continuación, se detalla un análisis de las competencias indirectas que, aunque no compiten directamente en el mismo nicho, podrían atraer a algunos de nuestros usuarios potenciales.

## **5.1 Competencias indirectas**

Para nuestra competencia indirecta podemos mencionar a Instagram representa una competencia indirecta para "Pet Cholito" ya que, aunque no está enfocada exclusivamente en el cuidado y adopción de mascotas, su extensa base de usuarios y su función de compartir contenido visual, crear perfiles para los usuarios y tener una gran comunidad pueden desviar la atención de nuestro público objetivo hacia otras comunidades de animales presentes en la plataforma.

## **5.2 Competencias directas**

Dentro del mercado de aplicaciones móviles enfocadas en mascotas, hemos identificado varias competencias directas, cada una con sus propias fortalezas y debilidades:

* **Fauna City**: Esta aplicación ofrece algunas funcionalidades similares a "Pet Cholito", pero carece de características clave como la opción de adopción de mascotas y la localización de veterinarias. Su enfoque está más limitado, lo que deja un espacio significativo para que "Pet Cholito" se diferencie con una oferta más completa.
* **Sosafe**: Aunque Sosafe es una aplicación de mapa donde los usuarios pueden reportar problemas y mascotas perdidas, su enfoque principal no son los animales sino diversos problemas sociales cotidianos. Por lo tanto, aunque es una competencia directa en términos de funcionalidad de mapa y reportes, su público objetivo es más amplio y menos especializado en mascotas.
* **PetTrace**: Similar a "Pet Cholito", PetTrace ofrece funcionalidades de rastreo de mascotas, pero también carece de opciones para la adopción y la visualización de veterinarias. La falta de estas características limitan su atractivo para usuarios que buscan una solución integral para el bienestar de sus mascotas.

Finalmente, aunque existen varias aplicaciones en el mercado que ofrecen algunas de las funcionalidades de "Pet Cholito", ninguna de ellas combina todas las características que nuestro producto ofrece. Con una interfaz intuitiva, completamente enfocada en las necesidades de los animales, y con funcionalidades únicas como la adopción, alertas en tiempo real, y la localización de puntos de interés relevantes como veterinarias y tiendas de accesorios para mascotas, "Pet Cholito" se posiciona como la opción preferida para los amantes de los animales que buscan una solución integral y especializada.

# 6. Evidencias

A continuación, se detallan las evidencias recopiladas a lo largo del desarrollo del proyecto "Pet Cholito". Estas evidencias reflejan los avances realizados en las diferentes etapas del proyecto, desde la conceptualización inicial hasta la implementación de una versión funcional de la aplicación.

1. **Tipo de evidencia: Avance  
   Nombre de la evidencia:** Mockups  
   **Descripción:** Diseños preliminares en Figma de la interfaz de usuario de la aplicación "Pet Cholito".  
   **Justificación:** Los mockups demuestran la conceptualización inicial de la aplicación y cómo se planea presentar las funcionalidades a los usuarios. Esto es crucial para evaluar la usabilidad y el diseño de la experiencia del usuario.
2. **Tipo de evidencia: Avance  
   Nombre de la evidencia:** Modelo de base de datos  
   **Descripción:** Esquema de la estructura de la base de datos que tendrá la aplicación.  
   **Justificación:** El modelo de base de datos es fundamental para entender cómo se organizará y almacenará la información de mascotas, usuarios y reportes. Esto refleja la competencia en gestión de bases de datos e información.
3. **Tipo de evidencia: Avance  
   Nombre de la evidencia:** Arquitectura de alto nivel  
   **Descripción:** Diagrama que muestra los componentes principales del sistema y cómo interactúan entre sí.  
   **Justificación:** La arquitectura de alto nivel demuestra la comprensión de cómo se estructurará la aplicación, incluyendo el frontend, backend y servicios en la nube. Esto es esencial para evaluar la competencia en diseño de software y arquitectura en la nube.
4. **Tipo de evidencia: Avance  
   Nombre de la evidencia:** Historias de usuario  
   **Descripción:** Documento que detalla las funcionalidades de la aplicación desde la perspectiva del usuario.  
   **Justificación:** Las historias de usuario ayudan a definir las funcionalidades requeridas por los usuarios finales, lo cual es fundamental para guiar el desarrollo centrado en el usuario y asegurar que se cumplan sus necesidades y expectativas.
5. **Tipo de evidencia: Final  
   Nombre de la evidencia:** Aplicación móvil  
   **Descripción:** Versión funcional de la aplicación "Pet Cholito" para dispositivos móviles en Android.  
   **Justificación:** La aplicación móvil es el producto final que integra todas las competencias desarrolladas durante el proyecto. Demuestra las habilidades en programación móvil, implementación de funcionalidades y medidas de seguridad.

# 7. Stack tecnológico

En el desarrollo de "Pet Cholito", hemos seleccionado cuidadosamente un stack tecnológico moderno y robusto que nos permita crear una aplicación móvil eficiente, escalable y de alto rendimiento. Nuestra elección de tecnologías está diseñada para abordar los desafíos específicos del proyecto, como la necesidad de desarrollo multiplataforma, la gestión de datos en tiempo real y la capacidad de manejar un gran número de usuarios concurrentes. A continuación, detallaremos cada componente de nuestro stack tecnológico y explicaremos por qué es crucial para el éxito de "Pet Cholito".

**React Native:** React Native es nuestro framework principal para el desarrollo de la aplicación móvil. Lo hemos elegido por varias razones clave:

1. **Desarrollo multiplataforma**: React Native nos permite desarrollar para iOS y Android simultáneamente, reduciendo significativamente el tiempo y costo de desarrollo.
2. **Rendimiento cercano al nativo**: Ofrece un rendimiento comparable a las aplicaciones nativas, crucial para una experiencia de usuario fluida.
3. **Gran comunidad y ecosistema**: Esto asegura un amplio soporte, numerosas bibliotecas y soluciones a problemas comunes.
4. **Facilidad de aprendizaje**: Para desarrolladores como nosotros estamos familiarizados con React,por lo que la curva de aprendizaje es menor.

**Firebase:** Firebase es nuestra plataforma backend as a service (BaaS). La hemos seleccionado por:

1. **Base de datos en tiempo real**: Esencial para las actualizaciones instantáneas de mascotas perdidas y encontradas.
2. **Autenticación de usuarios**: Proporciona un sistema seguro y fácil de implementar para el registro y login de usuarios.
3. *Almacenamiento en la nube*: Ideal para guardar imágenes de mascotas y otros archivos multimedia.
4. **Notificaciones push**: Crucial para alertar a los usuarios sobre mascotas perdidas en su área.
5. **Analítica**: Nos ayudará a entender el comportamiento de los usuarios y mejorar la aplicación.

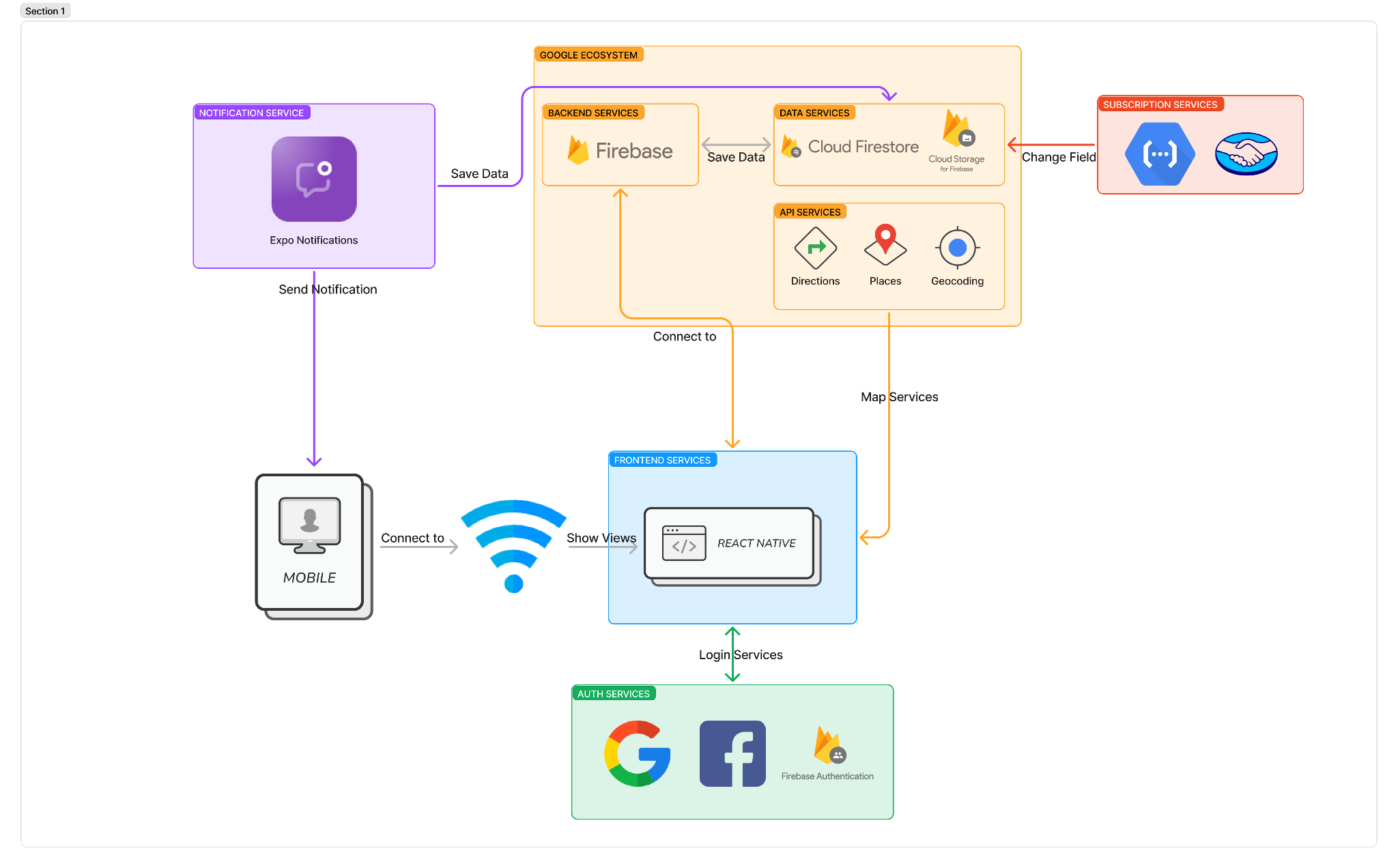
**Google Cloud Platform:** Utilizamos Google Cloud Platform (GCP) para servicios adicionales y escalabilidad:

1. **Cloud Storage**: Como complemento al almacenamiento de Firebase para archivos más grandes o menos frecuentemente accedidos.
2. **Maps API**: Para la funcionalidad de geolocalización y mapeo de mascotas perdidas y encontradas.

**Tecnologías adicionales:**

1. **Expo:** Plataforma utilizada para el desarrollo de aplicaciones móviles en React Native. Expo simplifica la creación, prueba y despliegue de la aplicación, permitiendo un desarrollo más rápido y eficiente.
2. **Figma:** Herramienta de diseño colaborativa utilizada para crear mockups y prototipos de la interfaz de usuario. Figma facilita la comunicación del equipo de diseño y desarrollo, asegurando que la interfaz cumpla con los requisitos funcionales y estéticos del proyecto.

## **8. Diagrama Arquitectura**



Este stack tecnológico nos proporciona una base sólida y flexible para desarrollar "Pet Cholito". Nos permite crear una aplicación robusta y escalable que puede manejar eficientemente las demandas de nuestros usuarios, desde la búsqueda en tiempo real de mascotas perdidas hasta la gestión de una comunidad activa de amantes de los animales. Además, estas tecnologías nos dan la capacidad de iterar rápidamente y adaptar la aplicación a las necesidades cambiantes de nuestros usuarios y el mercado.

# 9. Plan de trabajo

A continuación se detalla el plan de trabajo para el proyecto "Pet Cholito", destacando las competencias o unidades de competencias, actividades/tareas específicas, sus descripciones, recursos necesarios, duración y responsables.

**Gestión de proyectos informáticos**

* **Nombre de la Actividad/Tarea:** Planificación de los sprints
* **Descripción:** Definir y organizar las tareas en sprints, estableciendo objetivos y plazos para cada iteración del desarrollo.
* **Recursos:** Software de gestión de proyectos, plantillas en Word, Excel, y Google Drive.
* **Duración de la actividad:** Continua (actualización semanal)
* **Responsable:** Cristhian Barreno, Nicolás Ruiz
* **Observaciones:** Posible dificultad para coordinar tiempos y prioridades. Facilitador: uso de herramientas de gestión de proyectos ágiles.

**Diseño y desarrollo de software**

* **Nombre de la Actividad/Tarea:** Documentación Necesaria
* **Descripción:** Crear y mantener la documentación del proyecto, incluyendo requerimientos, especificaciones técnicas y manuales de usuario.
* **Recursos:** Herramientas de documentación como Word.
* **Duración de la actividad:** 4 semanas
* **Responsable:** Cristhian Barreno, Nicolás Ruiz
* **Observaciones:** Dificultad para mantener la documentación actualizada. Facilitador: usar plantillas predefinidas.

**Diseño de interfaces de usuario**

* **Nombre de la Actividad/Tarea:** Diseño de Mockups
* **Descripción:** Crear prototipos visuales de la interfaz de usuario de la aplicación "Pet Cholito".
* **Recursos:** Software de diseño (ej. Figma).
* **Duración de la actividad:** Continua (actualización semanal)
* **Responsable:** Cristhian Barreno, Nicolás Ruiz
* **Observaciones:** Dificultad para balancear estética y funcionalidad. Facilitador: feedback temprano de potenciales usuarios.

**Diseño y desarrollo de software**

* **Nombre de la Actividad/Tarea:** Validación de vistas
* **Descripción:** Revisar y probar los diseños de interfaz de usuario antes de la implementación.
* **Recursos:** Herramientas de revisión de diseño como encuestas en las redes sociales o con personas cercanas y reuniones de equipo.
* **Duración de la actividad:** Continua (actualización semanal)
* **Responsable:** Cristhian Barreno, Nicolás Ruiz
* **Observaciones:** Dificultad para conciliar diferentes opiniones. Facilitador: establecer criterios claros de validación.

**Programación en múltiples lenguajes**

* **Nombre de la Actividad/Tarea:** Programación de vistas
* **Descripción:** Implementar las interfaces de usuario diseñadas en el código de la aplicación móvil.
* **Recursos:** IDE de desarrollo móvil (Visual Studio Code), lenguajes de programación (JavaScript, React Native).
* **Duración de la actividad:** Continua (actualización semanal)
* **Responsable:** Cristhian Barreno, Nicolás Ruiz
* **Observaciones:** Dificultad para asegurar compatibilidad con diferentes dispositivos. Facilitador: uso de frameworks de desarrollo multiplataforma.

**Gestión de bases de datos e información**

* **Nombre de la Actividad/Tarea:** Integrar base de datos
* **Descripción:** Conectar la interfaz de usuario con la base de datos, implementando las operaciones CRUD necesarias.
* **Recursos:** Sistema de gestión de bases de datos (Firebase).
* **Duración de la actividad:** Continua (actualización semanal)
* **Responsable:** Cristhian Barreno, Nicolás Ruiz
* **Observaciones:** Dificultad para asegurar la eficiencia de las consultas. Facilitador: uso de Firebase para simplificar la interacción con la base de datos.

**Diseño y desarrollo de software, Ciberseguridad**

* **Nombre de la Actividad/Tarea:** Pruebas de Funcionamiento
* **Descripción:** Realizar pruebas unitarias, de integración y de seguridad para asegurar el correcto funcionamiento de la aplicación.
* **Recursos:** Herramientas de testing, frameworks de pruebas automatizadas.
* **Duración de la actividad:** 4 semanas
* **Responsable:** Cristhian Barreno, Nicolás Ruiz
* **Observaciones:** Dificultad para cubrir todos los casos de uso posibles. Facilitador: implementar metodologías de desarrollo guiado por pruebas.

# 10. Historias de Usuario

En el desarrollo de Pet Cholito, las historias de usuario son fundamentales para definir las funcionalidades desde la perspectiva del usuario final. Estas historias capturan las necesidades de los usuarios, lo que nos permite priorizar y desarrollar de manera eficiente las características más relevantes de la aplicación.

Para ver la plantilla de las historias de usuario utilizadas en este proyecto, puedes acceder al siguiente enlace: [Plantilla Historias Usuario.xls.xlsx](https://docs.google.com/spreadsheets/d/1sJOFvAOZiDtD6Ew1gxmGP1McttxDjC8S/edit?usp=sharing&ouid=108715306430008885512&rtpof=true&sd=true)

# 11. Base de Datos

La base de datos de "Pet Cholito" está diseñada para almacenar información sobre usuarios, perfiles de mascotas, publicaciones, seguidores y alertas. A continuación, se describe cada una de las tablas y las relaciones entre ellas.

## Estructura de la Base de Datos:

1. **users:** Esta tabla almacena la información de los usuarios registrados.

* id: Identificador único de usuario.
* displayName: Nombre visible del usuario.
* fcmToken: Token para envío de notificaciones.
* email: Correo electrónico del usuario.

1. **pet\_profiles:** Cada usuario puede tener uno o más perfiles de mascotas.

* id: Identificador único del perfil de la mascota.
* name: Nombre de la mascota.
* user\_name: Nombre de usuario de la mascota.
* image\_profile: URL de la imagen de perfil de la mascota.
* bio: Breve descripción del perfil.
* created\_at: Fecha de creación del perfil.
* updated\_at: Fecha de la última actualización del perfil.
* selectedProfile: Indica si es el perfil seleccionado.
* isVisible: Controla la visibilidad del perfil.
* user: ID del usuario dueño del perfil (relación con users).

1. **posts:** Los perfiles de mascotas pueden realizar publicaciones.

* id: Identificador único de la publicación.
* pet\_profile\_id: Relación con el perfil de mascota que creó la publicación. (relación con pet\_profiles).
* user: Relación el usuario dueño de la mascota (relación con users).
* images: Imágenes asociadas a la publicación.
* description: Texto de la publicación.
* postType: Tipo de publicación.
* created\_at: Fecha de creación de la publicación.

1. **followers:** Los perfiles de mascotas pueden seguir y ser seguidos por otros perfiles

* id: Identificador único del seguimiento.
* followedId: ID del perfil que está siendo seguido.
* followProfileId: ID del perfil que está siguiendo.
* followedAt: Fecha en la que se realizó el seguimiento.

1. **alerts:** Se utiliza para crear alertas relacionadas con mascotas, como pérdidas o adopciones

* id: Identificador único de la alerta.
* selectedAnimal: Tipo de animal en la alerta.
* name: Nombre del animal.
* breed: Raza del animal.
* age: Edad del animal.
* chip: Indica si tiene microchip.
* sex: Sexo del animal.
* description: Descripción de la alerta.
* images: Imágenes relacionadas con la alerta.
* typeAlert: Tipo de alerta (pérdida, adopción).
* date: Fecha de la alerta.
* location: Ubicación de la alerta.
* coordinates: Coordenadas exactas del lugar.
* user: Relación con el usuario que creó la alerta (relación con users).

1. **posts:** Se utiliza para la creación de posts, relacionado con los pet\_profiles

* id: Identificador único del post.
* commentsCount: Cantidad de comentarios.
* pet\_profiles\_id: id del pet\_profile dueño del post(relación con pet\_profiles).
* user: Relación con el usuario que creó la alerta (relación con users).
* images: Imágenes relacionadas con la alerta.
* description: Descripción de la alerta.
* pet\_username: Sexo del animal.
* postType: Descripción de la alerta.
* created\_at: Imágenes relacionadas con la alerta.
* commentsEnabled: Tipo de alerta (pérdida, adopción).
* likes: Fecha de la alerta.
* likesCount: Ubicación de la alerta.

1. **comments**: Almacena los comentarios realizados por los usuarios en los posts

* id: Identificador único del comentario.
* post\_id: Relación con el post al que pertenece (relación con posts).
* petId: Perfil de la mascota que realizó el comentario (relación con pet\_profiles).
* petName: Nombre de usuario del pet\_profiles.
* commentText: Contenido del comentario.
* created\_at: Fecha y hora de creación del comentario.
* petImage:URL de la imagen de perfil de la mascota.

1. **replies:** Esta tabla almacena las respuestas asociadas a los comentarios.

* id: Identificador único de la respuesta.
* comment\_id: Relación con el comentario al que pertenece (relación con comments).
* created\_at: Fecha y hora de creación del comentario.
* likes: Lista de identificadores de usuarios que dieron "like" a la respuesta.
* likesCount: Número total de "likes" de la respuesta.
* postId:Relación con el post asociado (relación con posts).
* petId: Perfil de la mascota que realizó el comentario (relación con pet\_profiles).
* petName: Nombre de la mascota que realizó la respuesta.
* petImage: URL o referencia de la imagen de la mascota que realizó la respuesta.
* replyText: Contenido del texto de la respuesta.

1. **notifications:** Almacena las notificaciones enviadas a los usuarios de la plataforma.

* id:Identificador único de la notificación.
* message:Contenido del mensaje de la notificación.
* petProfile:Identificador del perfil de mascota relacionado (si aplica).
* relatedId: Identificador relacionado con el contexto de la notificación (puede ser un post, comentario, etc.).
* userId:Identificador del usuario asociado a la notificación.
* type: Tipo de notificación (por ejemplo, "like", "comment", "follow").
* timestamp: Fecha y hora de creación de la notificación.
* user:Nombre o identificador del usuario que generó la notificación.

1. messages: Almacena los mensajes enviados entre usuarios en la plataforma.

* id: Identificador único del mensaje.
* txt: Contenido del mensaje.
* createdAt: Fecha y hora de creación del mensaje.
* mediaUrl: URL del contenido multimedia adjunto al mensaje (si aplica).
* messageType: Tipo de mensaje (por ejemplo, texto, imagen, archivo, etc.).
* user: Mapa de datos del usuario que envió el mensaje:
* user\_id:Identificador del usuario.
* user\_avatar: URL del avatar del usuario.
* user\_name: Nombre del usuario.

1. conversations: Almacena la información de las conversaciones entre usuarios o perfiles de mascotas.

* id: Identificador único de la conversación.
* messages: Mapa que contiene mensajes relacionados con esta conversación.
* avatars: Mapa de los avatares de los participantes.
* lastMessage: Contenido del último mensaje enviado en la conversación.
* lastMessageTime: Fecha y hora del último mensaje enviado.
* participantNames: Lista de nombres de los participantes de la conversación.
* participants: Lista de identificadores de los participantes.
* conversationId: Identificador único para la conversación.
* location:Ubicación relacionada con la conversación (si aplica).
* namePet: Nombre de la mascota (si la conversación está asociada a un perfil de mascota).

1. **partners**:Esta tabla almacena la información de los servicios ofrecidos por los socios.

* id: Identificador único del servicio.
* name: Nombre del servicio.
* partner\_ids: Lista de identificadores de socios (partners) que ofrecen este servicio.

1. **services**: Esta tabla almacena la información de los socios.

* id: Identificador único del socio.
* about: Descripción o información general sobre el socio.
* coordinates: Mapa que contiene las coordenadas exactas de la ubicación del socio.
* images: Lista de URLs de imágenes asociadas al socio.
* name: Nombre del socio.
* phone: Número de teléfono del socio.
* schedule: Mapa que define los horarios de disponibilidad del socio, incluyendo días de la semana y horas de apertura/cierre.
* valorations: Puntuación o valoración promedio del socio, basada en las reseñas de los usuarios.
* website: URL del sitio web del socio.

## Relación entre Tablas

**users ↔ pet\_profiles:** Un usuario puede tener múltiples perfiles de mascotas, lo que establece una relación de uno a muchos entre users y pet\_profiles.

**pet\_profiles ↔ posts:** Un perfil de mascota puede tener múltiples publicaciones, lo que establece una relación de uno a muchos entre pet\_profiles y posts.

**followers ↔ pet\_profiles:** Los perfiles de mascotas pueden seguirse entre ellos. Esta es una relación de uno a muchos, donde un perfil puede seguir a varios otros, pero un registro en followers indica un seguimiento entre dos perfiles.

**users ↔ alerts:** Un usuario puede crear varias alertas, lo que establece una relación de uno a muchos entre users y alerts.

**users ↔ pet\_profiles:** Un usuario puede tener múltiples perfiles de mascotas, pero cada perfil de mascota pertenece a un único usuario.

**pet\_profiles ↔ posts:** Cada perfil de mascota puede tener múltiples publicaciones, pero cada publicación pertenece a un único perfil.

**followers ↔ pet\_profiles:** Los perfiles de mascotas pueden seguir a otros perfiles. La relación se gestiona mediante una tabla intermedia llamada followers, que registra el vínculo entre un seguidor y un seguido.

**posts ↔ comments:** Un post puede tener múltiples comentarios, pero cada comentario pertenece a un único post.

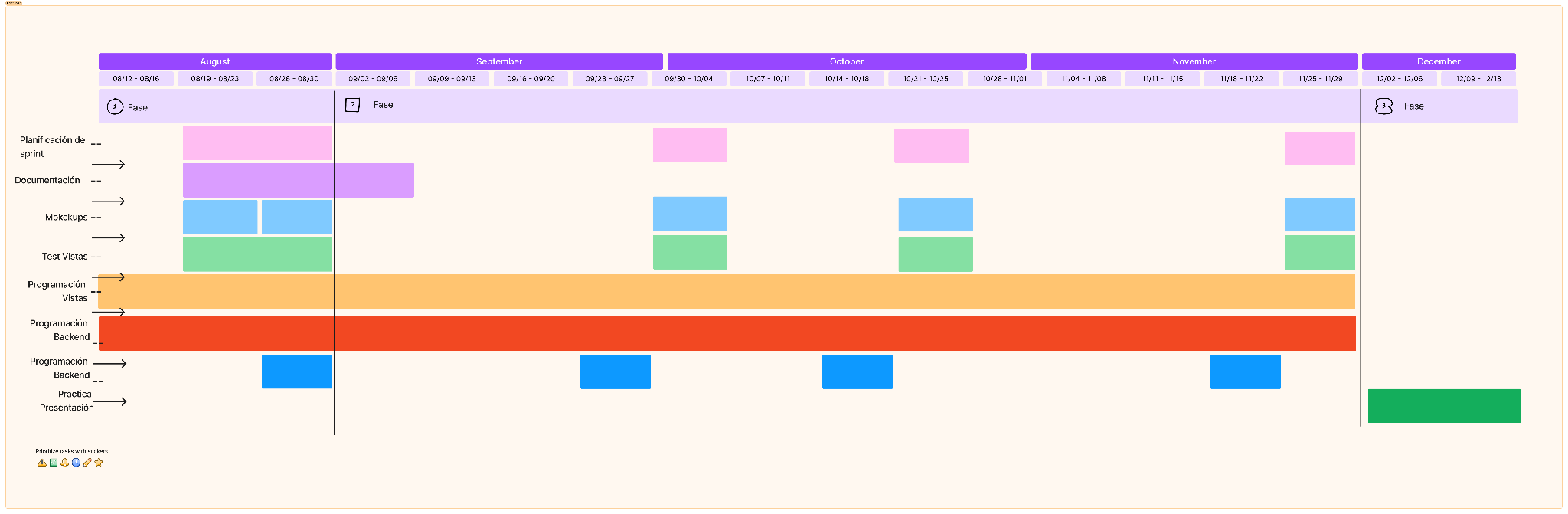
**comments ↔ replies:** Cada comentario puede tener múltiples respuestas, pero cada respuesta pertenece a un único comentario.

**partners ↔ services:**  Cada socio (partners) puede ofrecer múltiples servicios (services), pero cada servicio pertenece únicamente a un socio.

## Diagrama de Base de Datos

# 12. Carta Gantt

La carta Gantt es una herramienta clave para la planificación y seguimiento del proyecto "Pet Cholito". Proporciona una representación visual del cronograma, mostrando las fechas de inicio y finalización de cada tarea, así como sus interdependencias. Esta herramienta facilita la coordinación y gestión eficiente del proyecto, asegurando que se cumplan los objetivos y plazos establecidos. A continuación se presenta la carta Gantt, que detalla el plan de trabajo y el progreso esperado para cada fase del proyecto.



Para una correcta visualización ingrese a este link: [Figma Carta gantt](https://www.figma.com/board/3TXOhdY1yHqUYUt4tkurIw/PetCholito?node-id=98-3580&node-type=SECTION&t=XfVGXvpdOQhHIXDQ-0)

# 13. Plan de Pruebas

El Plan de Prueba**s** para el proyecto “Pet Cholito” detalla las estrategias, objetivos y cronogramas para garantizar que todas las funcionalidades de la aplicación se validen correctamente. Este documento establece los lineamientos necesarios para la planificación, ejecución y seguimiento de las pruebas.

Para consultar el Plan de Pruebas completo, accede al siguiente enlace:

[Plantilla de Plan de Pruebas de Software.docx](https://docs.google.com/document/d/1YGxwl35WMICo0hJCZu9SchHH7ij4hgbo/edit)

# 14. Casos de Prueba

En esta etapa del proyecto “Pet Cholito”, se han definido varios artefactos que nos permiten determinar los casos de prueba que serán aplicados para garantizar la funcionalidad correcta de las diferentes características de la aplicación.

Cada caso de prueba describe un escenario específico, enfocándose en validar la funcionalidad de acuerdo a los criterios de aceptación de las historias de usuario previamente definidas.

Para consultar los Casos de Pruebas, accede al siguiente enlace:

[Casos de Prueba](https://docs.google.com/spreadsheets/d/10SATr_xciJoDj1x4f7HXIxQa-NI2UkcG/edit?rtpof=true)

## 13.1 Tipos de Pruebas

Dentro del QA existen muchos tipos de pruebas que se utilizan para validar que una aplicación o sistema cumpla con un estándar de calidad y funcione correctamente.

Algunas de las pruebas que existen son las siguientes:

1. **Pruebas Funcionales**:Validan que el software funcione de acuerdo con los requisitos especificados.
2. **Pruebas No Funcionales**: Evalúan aspectos del sistema que no están directamente relacionados con las funcionalidades, como rendimiento, usabilidad y seguridad.
3. **Pruebas de Regresión**: Se realizan después de cambios en el código para asegurar que las funcionalidades previas siguen funcionando correctamente.
4. **Pruebas de Caja Negra**: Pruebas sin tener conocimiento interno del sistema; solo prueban las entradas y salidas sin preocuparse de la lógica interna.
5. **Pruebas de Caja Blanca**: Pruebas con acceso al código. Pueden realizar pruebas que involucren el flujo de datos, ciclos, y estructura de control.
6. **Pruebas de Aceptación**: Son pruebas realizadas por el usuario final para asegurarse de que el sistema cumple con los criterios de aceptación definidos.

Para los casos de prueba de Pet Cholito se realizaron las siguientes pruebas:

* **Prueba de funcionalidad**: Para verificar si funciona correctamente el requerimiento.
* **Pruebas de Integración**: Para verificar la integridad de servicios google.
* **Prueba de interfaz de usuario**: Verificar que los elementos visuales de la aplicación sean intuitivos, estén bien organizados y funcionen correctamente.
* **Prueba de Manejo de Errores**: Verificar que la aplicación maneje adecuadamente los errores y muestre mensajes de error.
* **Prueba de Restricción de Datos**: Asegurarse de que la aplicación límite los datos según las reglas definidas

[Plan de Plan de Pruebas](https://docs.google.com/document/d/1YGxwl35WMICo0hJCZu9SchHH7ij4hgbo/edit)

## 13.2 KPIs Pruebas

Los KPIs para pruebas nos permiten medir la efectividad de las pruebas al verificar si los requerimientos han sido cubiertos, si los casos de prueba se ejecutan correctamente y si se detectan errores graves. Manteniendo estos indicadores en niveles óptimos, se busca asegurar la calidad y funcionalidad del sistema.

| Objetivo | Asegurar que al menos el 90% de los requerimientos se hayan testeado mediante casos de prueba y se hayan aprobado, garantizando que las funcionalidades importantes del sistema sean evaluadas. |
| --- | --- |
| KPI | Porcentaje de requerimientos aprobados |
| Fórmula | (Número de Requerimientos Aprobados/ Número Total de Requerimientos) \* 100 |
| Frecuencia | Al final de cada sprint |

| Objetivo | Asegurar que el sistema cumpla con los requisitos de seguridad mediante pruebas de vulnerabilidad y penetración, reduciendo el riesgo de brechas de seguridad. |
| --- | --- |
| KPI | Porcentaje de Pruebas de Seguridad Completadas |
| Fórmula | (Número de Pruebas de Seguridad Ejecutadas / Número Total de Pruebas de Seguridad Planificadas) \* 100 |
| Frecuencia | Al final de cada sprint |

| Objetivo | Minimizar la cantidad de errores críticos que se detectan en la fase de pruebas para asegurar la estabilidad y calidad de la aplicación |
| --- | --- |
| KPI | Tasa de Errores Graves |
| Fórmula | (Número de Errores Graves Detectados / Número Total de Casos de Prueba Ejecutados) \* 100 |
| Frecuencia | Al final de cada sprint |

# 15. Seguridad

La seguridad de Pet Cholito se ha abordado mediante una combinación de prácticas y herramientas que garantizan la protección de los datos de los usuarios y la integridad de la aplicación. Para ello, se implementaron las siguientes medidas:

1. **Uso de Google Firebase**: La aplicación utiliza Firebase como sistema backend, lo que permite aprovechar la infraestructura de seguridad de Google. Firebase incluye autenticación segura y una capa de protección avanzada para manejar los datos de usuario, facilitando una integración segura con otras APIs y servicios en la nube.
2. **Encriptación de coordenadas en Alertas**: En el proceso de creación de alertas, las coordenadas de ubicación de la mascota se encriptan, protegiendo así la información sensible y evitando que terceros accedan a datos de ubicación sin autorización.
3. **Privacidad de Ubicación del Dispositivo**: La ubicación del dispositivo del usuario no se almacena en la aplicación, respetando así la privacidad de los usuarios y minimizando riesgos de exposición de datos sensibles. Únicamente se almacenan los datos estrictamente necesarios para el funcionamiento de las alertas de mascotas.
4. **Permisos y Autorizaciones**: La aplicación solicita únicamente los permisos necesarios para su operación, como acceso a la ubicación del dispositivo, notificaciones y archivos multimedia. Estos permisos se solicitan al usuario en el momento adecuado y con una descripción clara de su propósito, asegurando así que los usuarios entiendan y acepten la recopilación y uso de sus datos.
5. **Control de Acceso mediante Firebase Security Rules:** Las reglas de seguridad de Firebase se configuran para restringir el acceso a la base de datos, permitiendo que solo los usuarios autenticados puedan acceder a la información relevante. Esto asegura que cada usuario pueda ver únicamente sus propios datos y no los de otros usuarios.
6. **Monitoreo y Mantenimiento Continuo:** A través de Firebase Performance Monitoring y Crashlytics, la aplicación supervisa posibles problemas de seguridad y rendimiento en tiempo real, lo que permite actuar rápidamente ante cualquier amenaza.

Por otra parte en metodología se planea usar la **OWASP Mobile Application Security Verification Standard (MASVS)** que es un estándar de verificación de seguridad específicamente diseñado para aplicaciones móviles que permite guiar y evaluar la seguridad de la aplicación móvil en áreas clave, garantizando que cumpla con las prácticas recomendadas.

**Aplicación del MASVS en *Pet Cholito***:

1. **Control de Autenticación y Gestión de Sesiones**: La autenticación de usuarios se implementa mediante Firebase Authentication, siguiendo las recomendaciones del MASVS para gestionar sesiones de usuario de manera segura y prevenir ataques de suplantación.
2. **Protección de Datos Sensibles**: En *Pet Cholito*, todos los datos sensibles, como coordenadas de ubicación y perfiles de usuario, se encriptan antes de almacenarse y en tránsito, cumpliendo con los requisitos del MASVS para proteger la información en dispositivos móviles y servidores en la nube.
3. **Revisión de Permisos en el Dispositivo**: La aplicación sigue el principio de "mínimos privilegios" al solicitar únicamente los permisos necesarios (como la ubicación y el acceso a archivos multimedia) y lo hace de forma transparente para el usuario, asegurando el cumplimiento de las prácticas recomendadas del MASVS.
4. **Evaluación de Dependencias de Terceros**: Todas las librerías y dependencias de *React Native* y Firebase utilizadas en el proyecto se revisan para evitar vulnerabilidades conocidas. También se llevan a cabo actualizaciones periódicas para mantener la seguridad de los componentes.
5. **Pruebas de Seguridad Automatizadas y Manuales**: Se realizan pruebas de seguridad, tanto automatizadas como manuales, usando herramientas específicas para detectar vulnerabilidades comunes en aplicaciones móviles. Esto incluye pruebas de inyección, validación de entradas y accesos no autorizados.

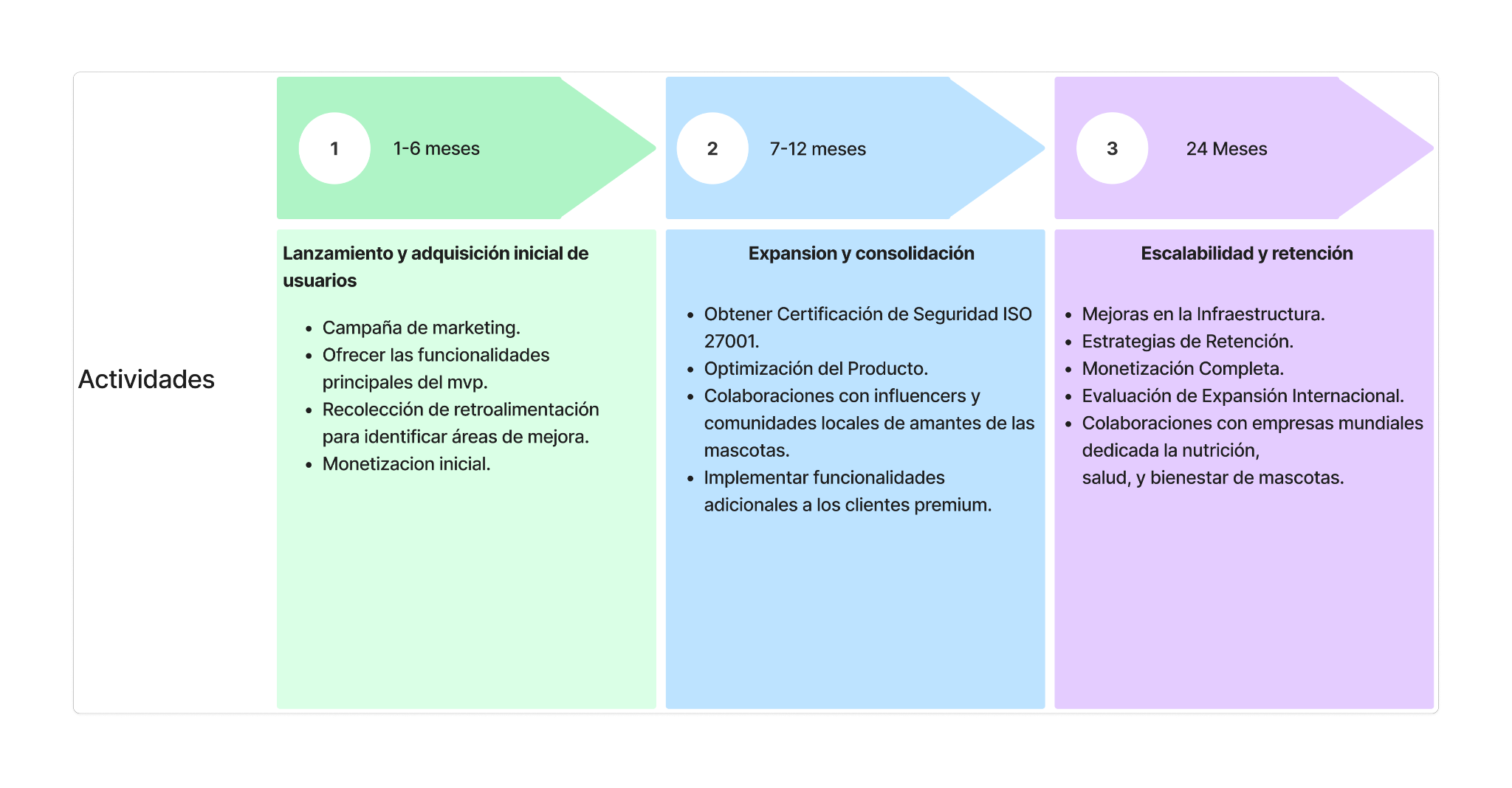
# 16. Flujo de caja Pet Cholito

El flujo de caja del proyecto proporciona una visión detallada de los ingresos y egresos mensuales, este muestra cómo se gestionan los recursos financieros a lo largo del tiempo. Este análisis es esencial para garantizar la viabilidad del proyecto y optimizar el uso de recursos.

Para ver el flujo de caja de Pet Cholito accede al siguiente enlace: [Caja Pet Cholito.xlsx](https://docs.google.com/spreadsheets/d/1qLMZivffcxAcYHBfsrLjPVnQ0kjtNmo3/edit?usp=drive_link&ouid=117012098905063077400&rtpof=true&sd=true)

# 17. Proyección de Escalabilidad de la Solución

La solución propuesta para "Pet Cholito" tiene un enfoque escalable, pensado para evolucionar con el tiempo y cubrir las necesidades crecientes de los usuarios y del mercado. A continuación, se detalla cómo se proyecta escalar la solución, las funcionalidades futuras planeadas y las razones detrás de las funcionalidades seleccionadas en el MVP.



En este caso que no puedas visualizar bien la imagen puedes entrar al flujo de caja de Pet Cholito y en el apartado de etapas se encuentra la imagen.

# 

# 18. Conclusión Individual

## **Nicolás Ruiz**

The development of the "Pet Cholito" project has been an incredibly rewarding and meaningful experience. I am very happy with the final result, as we not only managed to create a highly scalable application but also developed a platform with the potential to help many people and pets if used correctly. This application not only fosters a sense of community among animal lovers but also has the ability to make a real difference by helping reunite lost pets with their owners and promoting animal welfare.

Throughout the process, this project has allowed me to grow both personally and professionally. React Native has become a tool I use almost daily, and the high-performance technologies we implemented, such as Firebase, expo, and others, proved to be essential in achieving our goals. While I acknowledge there are areas in the application’s logic that could be improved, the overall outcome is highly satisfying.

I am proud to have been a part of this initiative and grateful for all the learning it provided along the way. Without a doubt, this project stands as a major accomplishment and a motivation to continue developing "Pet Cholito"

## **Cristhian Barreno**

For me, Pet Cholito is not just another project; it is a lifelong aspiration born from a concern I have carried with me for a long time. In the place where I live, there are many pets without owners, living in street conditions. This always made me reflect on how I could help them. By combining the knowledge I enjoy learning and my passion for technology, the idea of Pet Cholito came to life.

This project represents my commitment not only to technological development but also to the welfare of animals. Thanks to the support of my teammate Nicolás Ruiz and the unconditional encouragement of my family, I was able to turn this dream into reality. We not only managed to create something valuable, but the process also allowed me to integrate and strengthen my technical skills in programming, project management, and personal growth.

Pet Cholito is not just a technological contribution; it is an example of how technology can serve as a tool to create a positive impact on the lives of others, including those who cannot speak for themselves, such as abandoned pets. I am deeply grateful for the opportunity to work on such a meaningful project, which reinforces my commitment to continue learning and creating solutions that improve our society.

# 

# 19. Reflexión

The project has been an enjoyable and highly educational experience. Working on "Pet Cholito" has provided us with invaluable insights into the full lifecycle of mobile application development, from defining requirements to designing the user interface and beyond. Each step has challenged us to think critically and creatively, allowing us to deepen our understanding of what it takes to build a functional, user-centered app from the ground up.