

# Proyecto Pawfinder

**Universidad Modelo**

**Proyectos VII**

**Alumnos:**

García Jiménez Marielly (100)

Rejón Ancona Edwin Andrés (100)

Villamil Canto José Alberto (100)

**Profesora**

Ana Bolio Ayora

04/11/2025

# ÍNDICE

<b>RESUMEN</b>	<b>3</b>
<b>INTRODUCCIÓN</b>	<b>4</b>
<b>ANTECEDENTES</b>	<b>5</b>
<b>DEFINICIÓN DEL PROBLEMA</b>	<b>6</b>
<b>OBJETIVOS</b>	<b>9</b>
General	9
Específicos	9
<b>METODOLOGÍA</b>	<b>10</b>
Objetivo específico 1	10
Objetivo específico 2	10
Objetivo específico 3	11
Objetivo específico 4	11
Objetivo específico 5	11
Objetivo específico 6	11
<b>DISEÑO CONCEPTUAL</b>	<b>13</b>
<b>PLAN DE TRABAJO</b>	<b>18</b>
COSTOS	20
PLAN DE NEGOCIOS	21
<b>CONCLUSIONES</b>	<b>22</b>
<b>REFERENCIAS</b>	<b>23</b>
<b>ANEXOS</b>	<b>24</b>

# RESUMEN

Uno de los problemas más frecuentes en la ciudad de Mérida, Yucatán, es la pérdida de mascotas domésticas. Cuando esto sucede, los dueños recurren a métodos tradicionales como imprimir anuncios en hojas o publicar en redes sociales, lo cual suele ser ineficiente, desorganizado y de difícil seguimiento. La información se dispersa, los anuncios se pierden entre el flujo constante de publicaciones y la posibilidad de reunir voluntarios o testigos disminuye considerablemente. Es aquí donde Pawfinder busca marcar la diferencia. Esta aplicación móvil permitirá a los usuarios crear un perfil personal, publicar anuncios de sus mascotas perdidas e incluir información detallada como nombre, tipo de animal, raza, sexo, color, edad, rasgos distintivos, zona de extravío y fotografías.

Los anuncios se mostrarán tanto en un tablero general como en un mapa interactivo que indicará con un pin la zona donde se perdió la mascota. A través de filtros avanzados —como tipo de animal, color o zona—, otros usuarios podrán buscar y visualizar fácilmente los reportes más cercanos. Asimismo, el sistema permitirá ofrecer recompensas económicas, fomentando la colaboración entre usuarios, rescatistas y voluntarios locales.

El desarrollo de Pawfinder se llevará a cabo mediante el framework Flutter y el lenguaje Dart, que permiten crear aplicaciones móviles nativas y eficientes tanto para Android. La base de datos y servicios de autenticación se gestionarán a través de Firebase, lo que facilitará la conexión en tiempo real entre los usuarios y el almacenamiento de imágenes e información.

Al finalizar el proyecto, se espera contar con una aplicación moderna, accesible y funcional que se convierta en una alternativa eficiente y confiable para la búsqueda de mascotas extraviadas en Mérida.

# INTRODUCCIÓN

Pawfinder es un proyecto que se desarrolla entre la tecnología móvil, la empatía social y la protección animal, abordando uno de los problemas cotidianos más sensibles: la pérdida de mascotas. En un entorno urbano como Mérida, Yucatán, donde el crecimiento poblacional y la movilidad constante dificultan la búsqueda de animales extraviados, el uso de herramientas tecnológicas se vuelve una necesidad. Las redes sociales y los anuncios físicos cumplen una función, pero carecen de estructura, filtros y seguimiento. Pawfinder surge para ofrecer una solución moderna, organizada y accesible que conecta a dueños, rescatistas y voluntarios dentro de una misma comunidad digital.

La tecnología es el corazón de este proyecto. El desarrollo se realizará con Flutter y Dart, lo que permitirá crear una aplicación nativa, rápida y compatible con los principales sistemas operativos móviles. A través de Firebase, se gestionarán funciones esenciales como la autenticación de usuarios, almacenamiento de imágenes, base de datos y notificaciones en tiempo real. Estas herramientas facilitarán una experiencia de uso ágil y eficiente, asegurando que cualquier persona pueda publicar, buscar o compartir información sobre una mascota perdida de forma intuitiva.

La usabilidad es uno de los ejes principales de Pawfinder. La interfaz está diseñada para que cualquier usuario pueda crear un anuncio, buscar en el tablón o explorar el mapa interactivo sin complicaciones. Los filtros de búsqueda permitirán localizar mascotas por tipo, color, raza, sexo o zona geográfica, mientras que el mapa mostrará marcadores visuales que ayudarán a rastrear en áreas específicas.

Desde el punto de vista social, Pawfinder busca fomentar una cultura de apoyo y colaboración entre los habitantes de Mérida. Al ofrecer un canal centralizado para reportar y buscar mascotas, la aplicación no solo agiliza los reencuentros, sino que también fortalece el sentido de comunidad, la empatía y la responsabilidad compartida hacia los animales.

Por otro lado, Pawfinder representa una alternativa económica frente a los métodos tradicionales. Al eliminar la necesidad de imprimir volantes o depender de la visibilidad de las redes sociales, los usuarios pueden difundir información de forma gratuita, inmediata y

efectiva. Así, el proyecto combina tecnología, accesibilidad y compromiso social para responder a una necesidad real en la vida cotidiana.

## ANTECEDENTES

Actualmente, ya existen aplicaciones y plataformas digitales que ofrecen funcionalidades similares a las que se busca implementar en Pawfinder, enfocadas en la localización de mascotas perdidas o el apoyo en su búsqueda. Sin embargo, la mayoría de ellas están orientadas a un público internacional o requieren de pagos por suscripción, lo que limita su accesibilidad y alcance en comunidades locales como Mérida, Yucatán.

**Pet Alert México**, según indica su página oficial

(<https://www.facebook.com/petalertmexico/>), funciona como una red social dedicada a difundir casos de mascotas perdidas y encontradas mediante publicaciones. Aunque su impacto en redes es notable, la información no se encuentra organizada ni centralizada, lo que dificulta su búsqueda por zona, tipo de animal o características físicas. Además, al depender de la visibilidad en Facebook, los reportes suelen perderse entre el flujo constante de contenido.

**PawBoost**, de acuerdo con lo que se menciona en su sitio web oficial

(<https://www.pawboost.com>), es una plataforma estadounidense que permite publicar anuncios de mascotas perdidas o encontradas. Su sistema cuenta con alertas a comunidades locales y refugios, pero su servicio premium tiene un costo mensual y su cobertura en México es limitada. Asimismo, su diseño y comunicación están completamente en inglés, lo que representa una barrera para algunos usuarios.

Por último, **Finding Rover** (<https://findingrover.com>) es una aplicación que utiliza reconocimiento facial para identificar perros perdidos a partir de fotografías. Si bien esta tecnología resulta innovadora, requiere de una base de datos extensa para funcionar correctamente y aún no cuenta con soporte total en el país. Además, su interfaz y funcionamiento dependen de una conexión constante y estable, lo que puede complicar su uso en zonas con baja cobertura.

# DEFINICIÓN DEL PROBLEMA

En el transcurso del año 2025, se ha observado un incremento notable en la cantidad de mascotas extraviadas en la ciudad de Mérida, Yucatán, especialmente en la zona noreste, una de las áreas con mayor desarrollo urbano en los últimos años. Esta expansión ha traído consigo un aumento en la movilidad de los habitantes y en el número de mascotas domésticas, lo que ha generado nuevos desafíos para su cuidado y localización en caso de extravío.

Actualmente, los métodos más comunes para buscar mascotas perdidas son la impresión de anuncios físicos (colocados en postes, parques o establecimientos locales) y las publicaciones en redes sociales. Sin embargo, ambos métodos presentan limitaciones importantes. Los anuncios impresos dependen del tránsito físico de las personas y tienden a perderse o deteriorarse con el tiempo, mientras que las publicaciones en redes sociales se pierden rápidamente entre el alto volumen de contenido, dificultando su visibilidad y seguimiento. Además, no existe una forma organizada ni centralizada de acceder a todos los reportes de mascotas extraviadas dentro de la ciudad.

Esta situación ha generado frustración entre los dueños, quienes muchas veces no saben por dónde comenzar su búsqueda o carecen de herramientas digitales adecuadas para difundir la información de manera efectiva. Del mismo modo, los voluntarios y rescatistas locales enfrentan dificultades para identificar rápidamente los casos más cercanos o recientes, lo que reduce la posibilidad de éxito en los reencuentros.

Ante esta problemática, se propone el desarrollo de una aplicación móvil que concentre en un solo espacio todos los reportes de mascotas perdidas y facilite su búsqueda mediante filtros y herramientas de geolocalización. Pawfinder permitirá a los usuarios publicar información detallada y precisa de sus mascotas extraviadas, acompañada de fotografías y de la ubicación estimada del extravío en un mapa interactivo. Esta plataforma busca optimizar la comunicación entre los ciudadanos y fomentar la colaboración comunitaria, incrementando así las posibilidades de que cada mascota vuelva a casa.

# JUSTIFICACIÓN

Este proyecto se desarrollará con el propósito de ofrecer una solución tecnológica moderna y accesible al problema de la pérdida de mascotas en la ciudad de Mérida, Yucatán. Pawfinder busca facilitar y agilizar el proceso de búsqueda, centralizando en una sola plataforma la información sobre animales extraviados y promoviendo la colaboración entre los habitantes de la ciudad.

La aplicación proporcionará diversos beneficios, entre ellos:

- **Mayor eficiencia en la búsqueda:** Al permitir publicar reportes de mascotas perdidas con información detallada (fotografías, características físicas y ubicación geográfica), los usuarios podrán identificar con rapidez coincidencias y obtener resultados más precisos que en los métodos tradicionales.
- **Accesibilidad y alcance inmediato:** A diferencia de los anuncios físicos o publicaciones en redes sociales, Pawfinder concentrará los reportes en un tablero digital y un mapa interactivo, facilitando que cualquier usuario con conexión a internet pueda consultar casos cercanos o recientes en cuestión de segundos.
- **Fomento de la colaboración ciudadana:** La aplicación promoverá el trabajo conjunto entre dueños, voluntarios y rescatistas, creando una red de apoyo comunitaria orientada al rescate animal.
- **Ahorro de tiempo y recursos:** Al eliminar la necesidad de imprimir carteles o depender del azar en redes sociales, Pawfinder permitirá una difusión más rápida, gratuita y ordenada, reduciendo el esfuerzo y los gastos involucrados en la búsqueda.
- **Conciencia y responsabilidad social:** Además de servir como herramienta práctica, la aplicación busca generar una cultura de cuidado responsable de los animales domésticos y sensibilizar a la población sobre la importancia de actuar ante los casos de extravío.

Este proyecto será de gran utilidad para familias, estudiantes, trabajadores y rescatistas locales que deseen contar con una herramienta confiable, moderna y fácil de usar para reportar o encontrar mascotas. Pawfinder es una oportunidad de generar un impacto social

positivo, al ofrecer la tecnología móvil como una herramienta a beneficio de los animales y sus dueños.



# OBJETIVOS

## General

Desarrollar una aplicación móvil intuitiva y funcional que permita a los usuarios publicar, buscar y localizar mascotas extraviadas en la ciudad de Mérida, Yucatán, mediante el uso de herramientas de geolocalización, filtros de búsqueda y un tablero centralizado de anuncios, fomentando así la colaboración ciudadana y la recuperación eficiente de los animales perdidos.

## Específicos

1. Implementar un sistema de registro y autenticación de usuarios que permita crear perfiles personales y gestionar sus publicaciones de mascotas extraviadas.
2. Desarrollar un módulo de publicación de anuncios que incluya información detallada sobre cada mascota, como tipo, raza, sexo, color, edad, rasgos distintivos, zona de extravío y fotografías.
3. Integrar un mapa interactivo que muestre los reportes de mascotas perdidas mediante pines de ubicación, facilitando la visualización geográfica de los casos.
4. Incorporar filtros de búsqueda en el tablón de anuncios que permitan a los usuarios localizar anuncios según criterios específicos, como tipo de animal, color, zona o fecha de extravío.
5. Diseñar una interfaz de usuario moderna, accesible y responsiva utilizando Flutter, priorizando la facilidad de uso y la eficiencia en la navegación.
6. Establecer la conexión con una base de datos en tiempo real mediante Firebase, para garantizar el almacenamiento y la actualización inmediata de la información.

# METODOLOGÍA

## Objetivo específico 1

- **Procedimiento**

- Diseñar y codificar las interfaces necesarias para la creación y autenticación de usuarios.
- Diseñar y crear las tablas en la base de datos para contener los datos de registro de los usuarios.
- Conectar las pantallas de login y de registro para que se comuniquen con la base de datos y puedan crear o leer datos de ella.

- **Herramientas**

- Figma para los diseños.
- Draw.io para el esquema de la base de datos.
- Dart y el Framework Flutter.
- Firebase para la base de datos.

## Objetivo específico 2

- **Procedimiento**

- Analizar y definir las funcionalidades requeridas para poder crear, eliminar y editar posts publicados en la app.
- Diseñar y codificar las pantallas necesarias para llevar a cabo el proceso de crear una publicación.
- Planear la estructura de las tablas y sus relaciones con los usuarios en la base de datos.
- Implementar las tablas en la base de datos para luego conectarlas a las pantallas de creación de publicación.

- **Herramientas**

- Figma para los diseños.
- Draw.io para el esquema de la base de datos.
- Dart y el Framework Flutter.
- Firebase para la base de datos.

## Objetivo específico 3

- **Procedimiento**
  - Configurar la API Key de Google Cloud.
  - Integrar las APIs de Google Maps en la aplicación.
  - Usar los datos en las tablas de anuncios para mostrar las localizaciones de los lugares dónde se requiere atención.
- **Herramientas**
  - Chrome Dev Tools.
  - Firebase.
  - Dart con Framework Flutter.

## Objetivo específico 4

- **Procedimiento**
  - Crear Widget en la pantalla de posts que contenga tags que el usuario puede seleccionar para filtrar las publicaciones en el tablero.
  - Implementar en el backend los queries que se harán a la base de datos para traer los posts filtrados.
- **Herramientas**
  - Firebase.
  - Dart con Framework Flutter.

## Objetivo específico 5

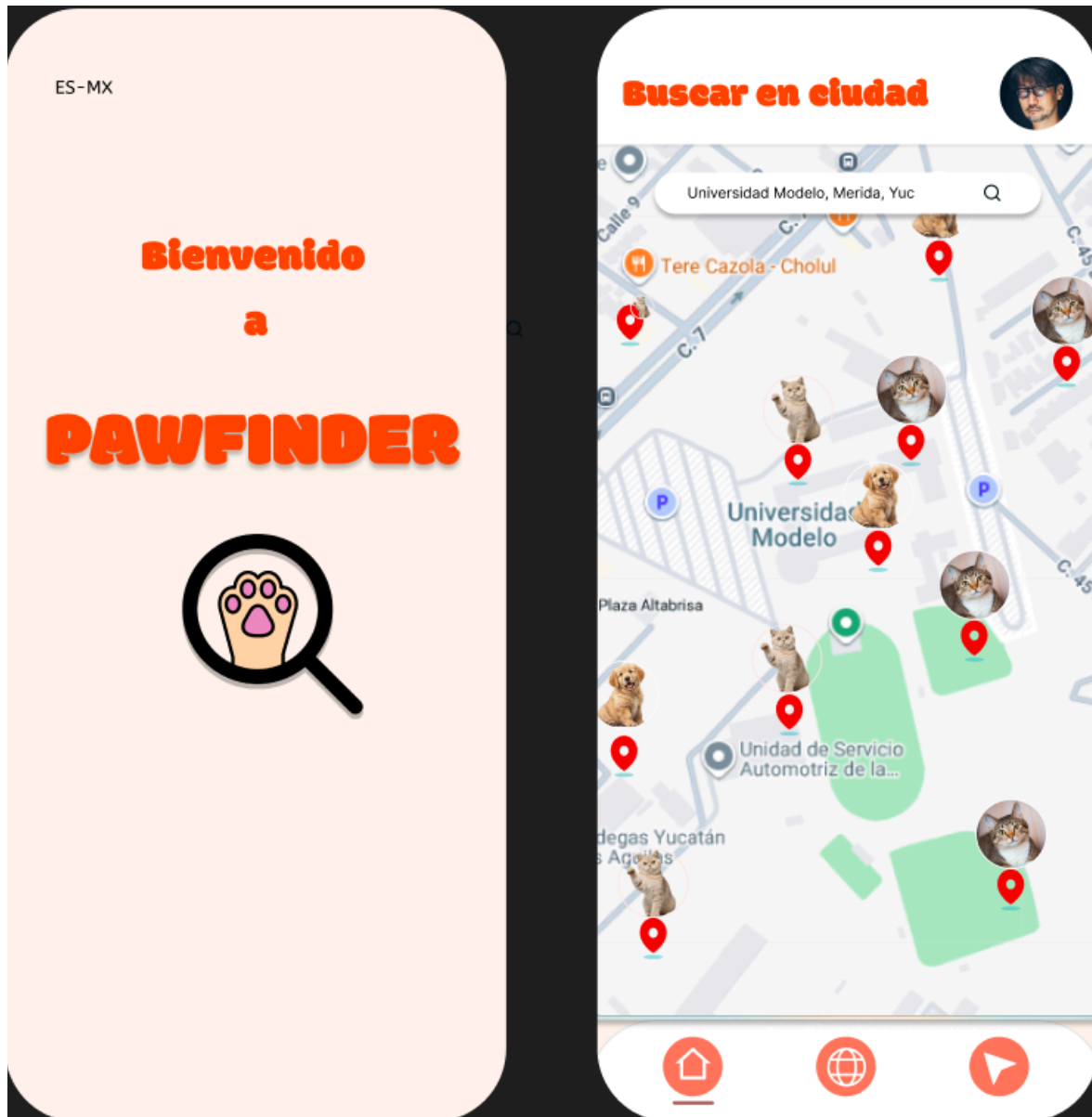
- **Procedimiento**
  - Realizar wireframes de las pantallas que contendrá la aplicación.
  - Utilizar las diversas herramientas de Figma para crear una interfaz elegante y fácil de navegar.
  - Crear una simulación de la navegación entre las diferentes pantallas.
- **Herramientas**
  - Figma.

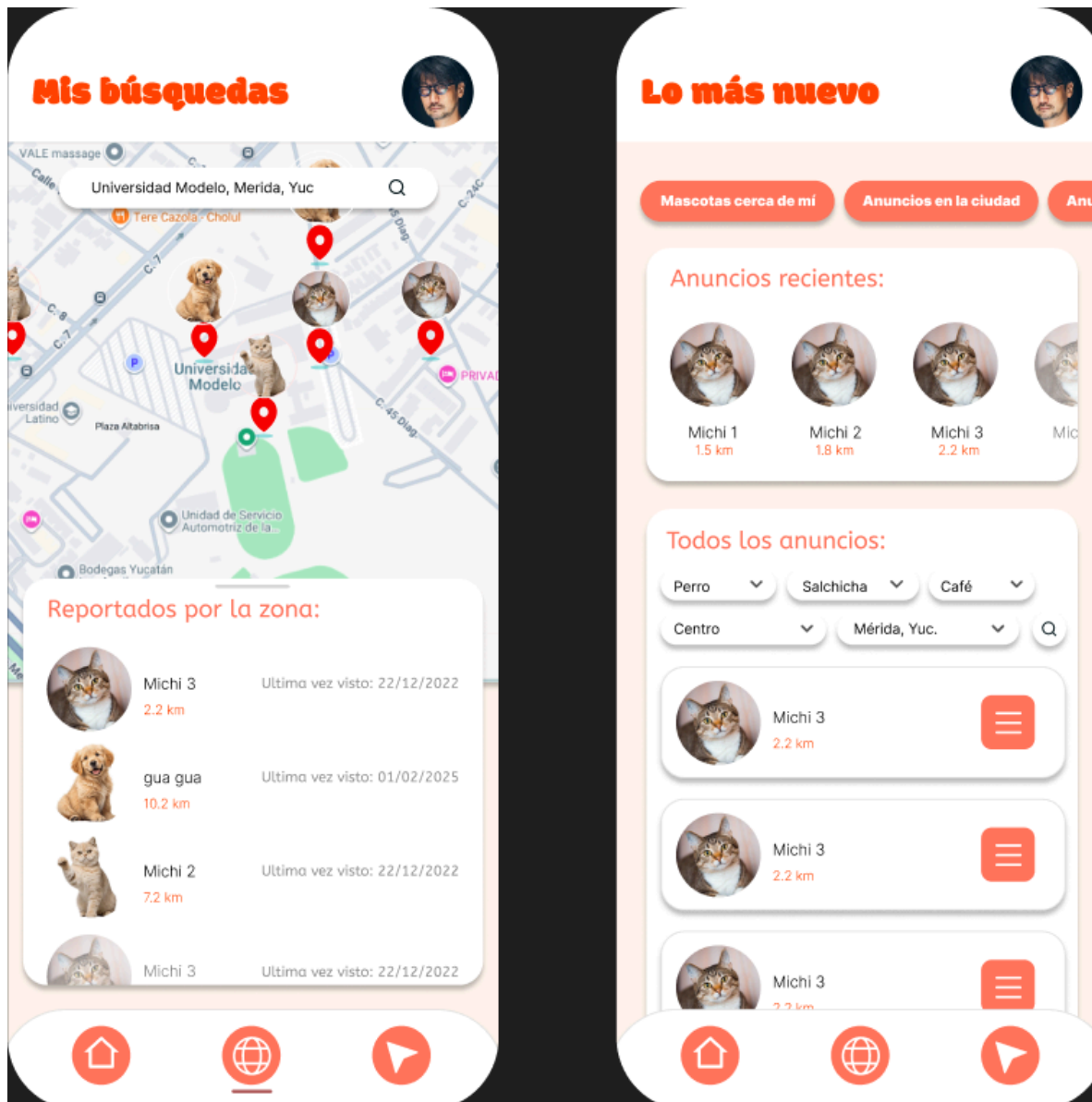
## Objetivo específico 6

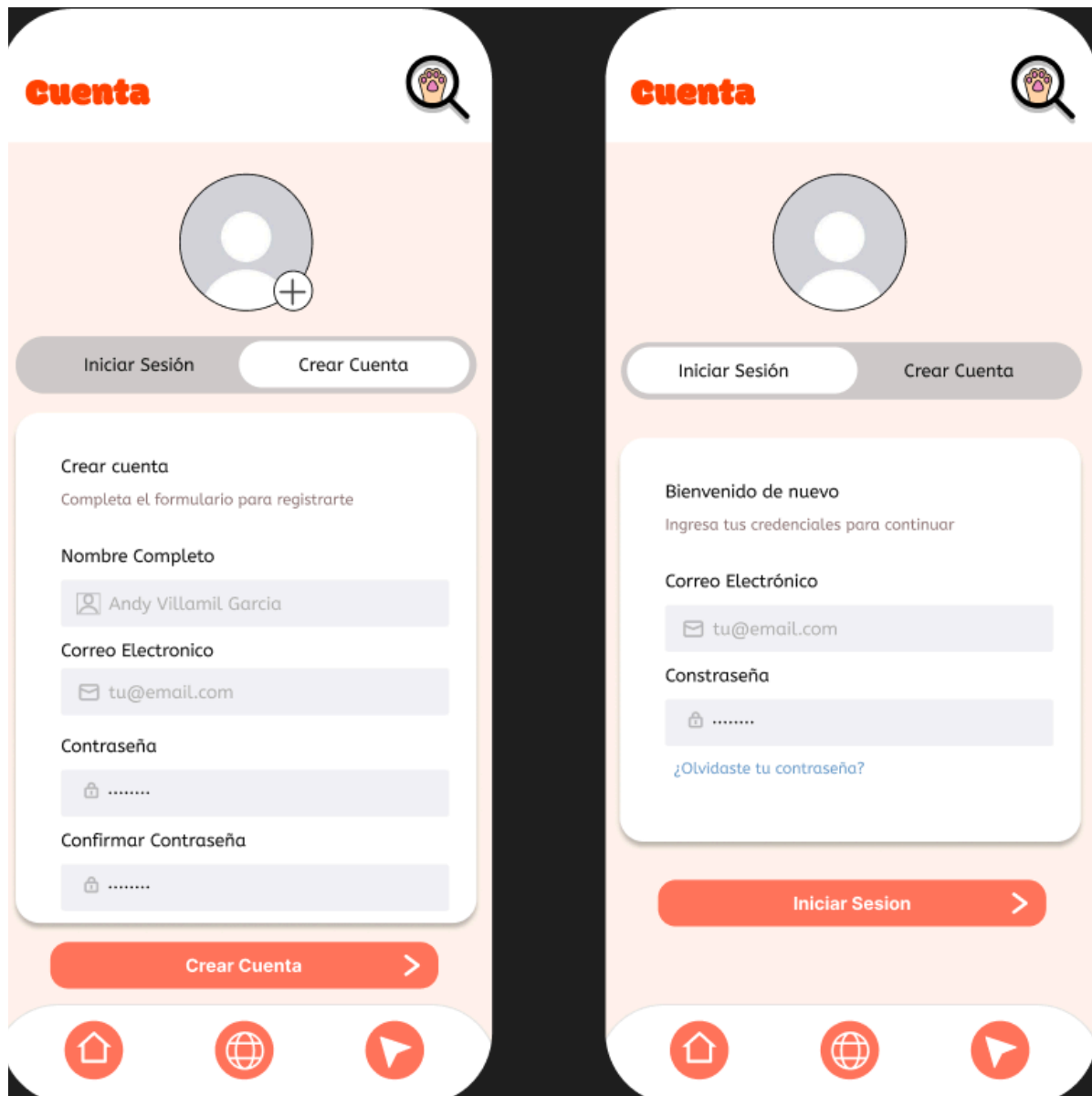
- **Procedimiento**

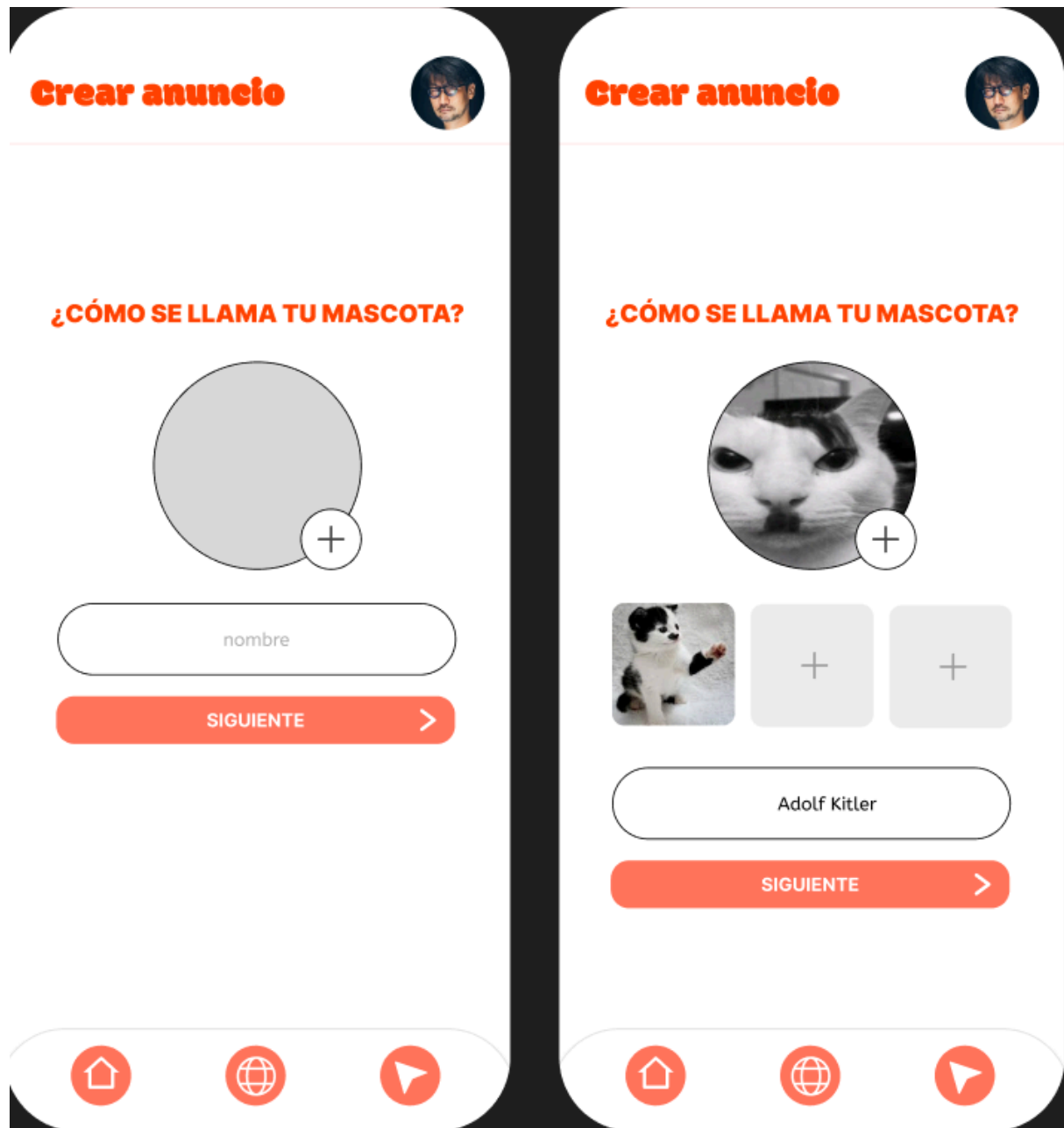
- Crear un proyecto de Firebase.
  - Obtener las credenciales para conectar Firebase con la app.
  - Diseño de la estructura de la base de datos y las relaciones de las tablas.
- **Herramientas**
  - Firebase.
  - Dart con Framework Flutter.

# DISEÑO CONCEPTUAL













## Crear anuncio





Nombre: Adolf Kitler

Tipo: Gato





Raza: Aria

Color: Blanco

**Detalles:**

Ejemplo: collar amarillo, ultima vez visto en el centro. Tiene un lunar en la cabeza.

**Galeria:**




SIGUIENTE

<

Adolf Kitler

>



¿Lo has visto?

Contactate al 9992339907

# PLAN DE TRABAJO

[illegible]

**Roles y responsabilidades**

Rol	Responsable
Líder de proyecto	Andrés Rejón
Responsable de diseño UI/UX	Andrés Rejón y Marielly García
Responsable de pruebas	Andrés Rejón, Marielly García y José Villamil
Responsable de implementación / DevOps	José Villamil
Responsable de documentación	Andrés Rejón, Marielly García y José Villamil

# COSTOS

Costos considerando gastos esenciales

Los gastos del proyecto Pawfinder se estiman tomando en cuenta los recursos necesarios para el desarrollo, mantenimiento y operación básica de la aplicación. Aunque muchas herramientas que se utilizarán son gratuitas (como Flutter, Firebase o Figma), existen costos relacionados con servicios, licencias, dominio y el tiempo de trabajo del equipo.

Gastos de administración por mes (en pesos mexicanos)

Gasto	Total al mes (MXN)
Licencia de GitHub	\$476.00
Dominio (www.pawfinder.mx)	\$350.00
Servidor Firebase/Cloud	\$1,200.00
Desgaste de computadoras (3 personas)	$\$300.00 \times 3 = \$900.00$
Nómina (3 personas, 80 h/mes, \$115/h como estudiantes)	\$27,600.00
<b>Total mensual aproximado</b>	<b>\$30,526.00</b>

**Tabla 1. Costos de desarrollo contando los gastos esenciales.**

## Publicidad

Se planea usar campañas en **redes sociales** (Facebook, Instagram y TikTok) enfocadas en asociaciones y comunidades locales de mascotas en Mérida.

El costo mensual estimado de publicidad es de **\$2,000.00 MXN**.

## Ganancia esperada (GE)

Se estima obtener al menos **el 50% de los costos mensuales** como ganancia proyectada, por lo tanto:

**GE = \$15,263.00 MXN mensuales.**

## Cálculo del costo total usando la fórmula:

Costo total =  $[GE \times (1 + IM) + GA + PU]$

Donde:

- **GA:** Gastos de administración = \$30,526.00
- **PU:** Publicidad = \$2,000.00
- **GE:** Ganancia esperada = \$15,263.00
- **IM:** Impuestos = 0.3

**Sustituyendo:**

Costo total =  $[15,263 \times (1 + 0.3) + 30,526 + 2,000]$

Costo total =  $[19,841.9 + 32,526]$

Costo total = **\$52,367.9 MXN mensuales**

Dado que el desarrollo del proyecto se estima en **10 meses**, el **costo total del proyecto** asciende aproximadamente a:

**\$523,679.00 MXN.**

**PLAN DE NEGOCIOS**

*Pawfinder* será una aplicación gratuita para los usuarios, y la recuperación de la inversión se realizará mediante **publicidad integrada y servicios premium** dentro de la app.

Los principales métodos de monetización incluyen:

- Publicidad local de veterinarias, tiendas de mascotas y refugios.
- Publicaciones destacadas de pago para los usuarios que deseen mayor visibilidad.
- Donaciones voluntarias para el mantenimiento del sistema.

De acuerdo con proyecciones de adopción en los primeros dos años, se espera alcanzar un promedio de 50,000 usuarios activos mensuales en Mérida y sus alrededores.

Con este nivel de tráfico, y aplicando un modelo básico de anuncios integrados, se podrían generar aproximadamente \$12,000 MXN mensuales, lo que permitiría recuperar la inversión inicial en un plazo de 4 a 5 años.

A medida que la aplicación se consolide y se expanda a otros estados, los ingresos aumentarán, permitiendo cubrir los costos operativos, mantenimiento y actualizaciones futuras.

# CONCLUSIONES

Respecto a la factibilidad técnica, el uso de tecnologías como Flutter, Dart, Firebase y Google Maps API garantiza una base sólida y moderna para el desarrollo del proyecto Pawfinder. Estas herramientas permiten crear aplicaciones móviles eficientes, escalables y compatibles con Android, lo cual facilita el despliegue y mantenimiento de la plataforma. Además, el equipo ha seleccionado herramientas complementarias como Figma, Draw.io y Visual Studio Code, que facilitarán el diseño, la organización y el trabajo colaborativo durante todas las etapas del desarrollo. El uso de Firebase, con su sistema de autenticación, base de datos en tiempo real y almacenamiento en la nube, contribuye a ofrecer un servicio seguro, confiable y de fácil gestión.

En cuanto a la factibilidad logística, Pawfinder responde a una necesidad social real: la pérdida frecuente de mascotas en la ciudad de Mérida. Su enfoque en conectar a dueños, rescatistas y voluntarios dentro de una sola aplicación crea una solución práctica y empática que puede integrarse fácilmente en la vida cotidiana de los usuarios. El hecho de que se trate de una app gratuita y de fácil acceso incrementa sus posibilidades de adopción y éxito, especialmente dentro de comunidades locales interesadas en la protección animal.

Desde la perspectiva económica, el proyecto es financieramente viable, ya que la mayoría de las herramientas utilizadas son de libre acceso o cuentan con planes gratuitos. Aunque el tiempo de recuperación de la inversión puede extenderse a varios años, existen oportunidades para implementar modelos de monetización sostenibles, como la publicidad local, servicios premium o publicaciones destacadas. Estos métodos permitirán mantener el funcionamiento de la aplicación a largo plazo sin comprometer su accesibilidad para los usuarios.

# REFERENCIAS

Firebase. (2025). *Firestore Documentation*. Google Developers. Recuperado de: <https://firebase.google.com/docs>

Flutter. (2025). *Build apps for any screen*. Google Flutter. Recuperado de: <https://flutter.dev>

Google Maps Platform. (2025). *Maps SDK for Android overview*. Google Developers. Recuperado de: <https://developers.google.com/maps/documentation/android-sdk>

Figma. (2025). *Collaborative interface design tool*. Recuperado de: <https://www.figma.com>

Organización Mundial de Sanidad Animal (OIE). (2023). *Bienestar animal y tenencia responsable de mascotas*. Recuperado de: <https://www.woah.org/es/>

# ANEXOS

En la siguiente imagen se muestra el código de la pantalla de la vista del mapa.

Aquí se seleccionan 3 lugares con el motivo de probar que funciona el marcador de Google Maps.

```
lib > map_screen.dart > _MapScreenState > _buildBottomNavBar
1  import 'package:flutter/material.dart';
2  import 'package:google_maps_flutter/google_maps_flutter.dart'; // Necesitarás este paquete
3  import 'package:pawfinder2/auth_screen.dart';
4
5  // Definimos los colores principales basados en tus imágenes
6  const Color kPrimaryColor = Color(0xFFFF2756D);
7  const Color kDarkGreyColor = Color(0xFF6C6C6C);
8
9  class MapScreen extends StatefulWidget {
10   const MapScreen({Key? key}) : super(key: key);
11
12   @override
13   _MapScreenState createState() => _MapScreenState();
14 }
15
16 class _MapScreenState extends State<MapScreen> {
17   GoogleMapController? mapController;
18
19   // Coordenadas para "Universidad Modelo, Mérida, Yuc"
20   static const LatLng _center = LatLng(21.0180, -89.5850); // Ajusta estas coordenadas si son más precisas para tu caso
21
22   // Creamos algunos marcadores de ejemplo
23   final Set<Marker> _markers = {
24     Marker(
25       markerId: MarkerId('university_model'),
26       position: _center,
27       infoWindow: InfoWindow(title: 'Universidad Modelo'),
28       icon: BitmapDescriptor.defaultMarkerWithHue(BitmapDescriptor.hueRed), // Marcador rojo para la ubicación principal
29     ), // Marker
30     // Agrega más marcadores para simular los pines con las imágenes de perrito
31     Marker(
32       markerId: MarkerId('location_1'),
33       position: LatLng(21.0195, -89.5870),
34       infoWindow: InfoWindow(title: 'Tere Cazola'),
35       icon: BitmapDescriptor.defaultMarkerWithHue(BitmapDescriptor.hueRed),
36     ), // Marker
37     Marker(
```

Lo siguiente es cargar el mapa en la pantalla.



```

16 class _MapScreenState extends State<MapScreen> {
54
55   @override
56   widget build(BuildContext context) {
57     return Scaffold(
58       appBar: AppBar(
59         backgroundColor: Colors.transparent, // Fondo transparente
60         elevation: 0,
61         titleSpacing: 0, // Elimina el espacio predeterminado del titulo
62         flexibleSpace: Container(
63           // Contenedor para el fondo superior del AppBar
64           decoration: const BoxDecoration(
65             color: Colors.white, // Color blanco para el fondo
66             borderRadius: BorderRadius.vertical(bottom: Radius.circular(20)), // Bordes redondeados en la parte inferior
67           ), // BoxDecoration
68         ), // Container
69         title: Padding(
70           padding: const EdgeInsets.symmetric(horizontal: 16.0),
71           child: Row(
72             mainAxisAlignment: MainAxisAlignment.spaceBetween,
73             children: [
74               Text(
75                 "Buscar en ciudad",
76                 style: TextStyle(
77                   color: Colors.primaryColor,
78                   fontWeight: FontWeight.bold,
79                   fontSize: 24,
80                   // Ajusta la fuente si es personalizada como en tu imagen
81                 ), // TextStyle
82             ), // Text
83             CircleAvatar(
84               radius: 20,
85               backgroundImage: NetworkImage(
86                 'https://i.imgur.com/your_profile_image.jpg'), // Reemplaza con una URL de imagen de perfil // NetworkImage
87               backgroundColor: Colors.grey[200],
88             ), // CircleAvatar

```

```

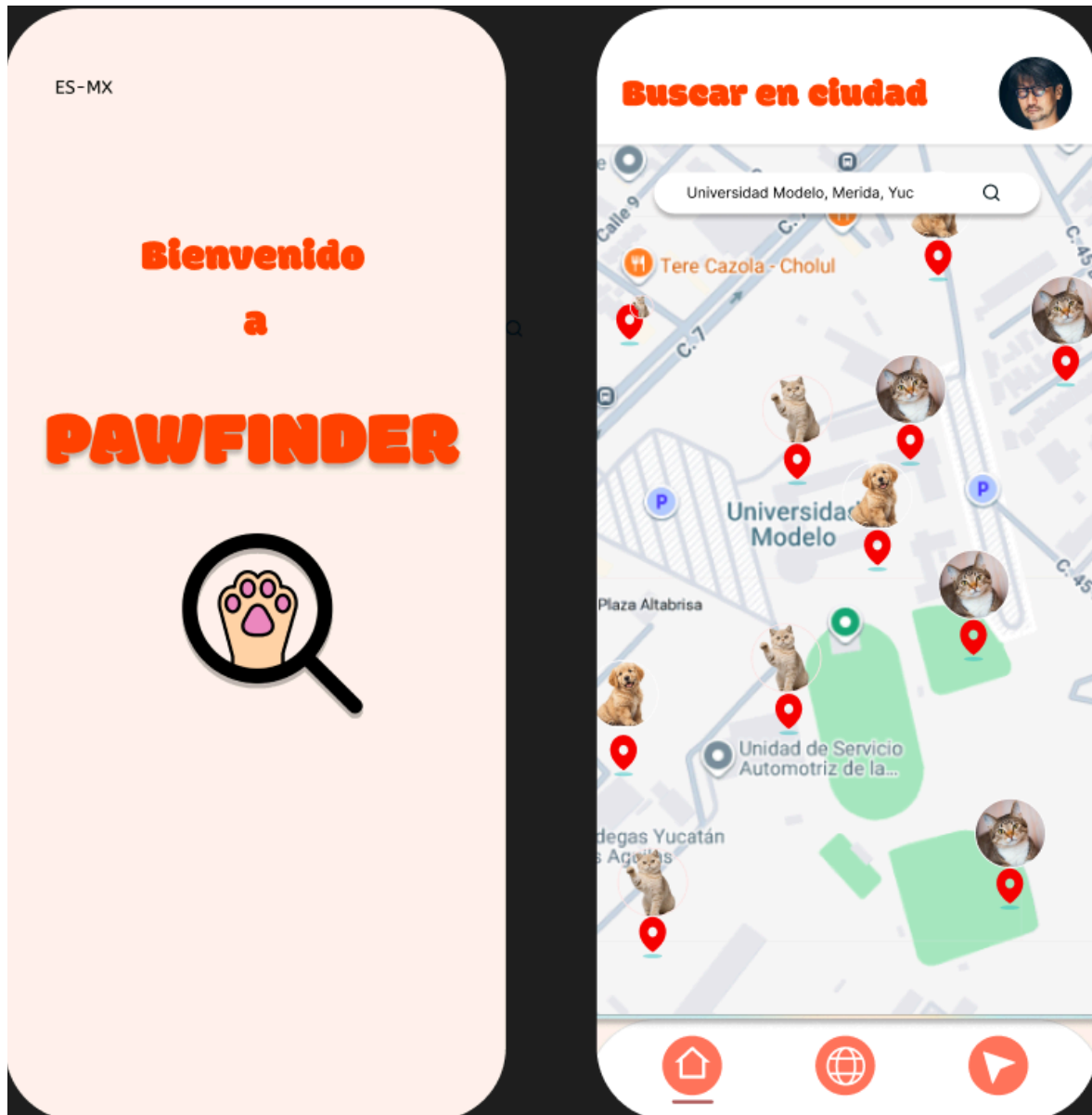
16 class _MapScreenState extends State<MapScreen> {
56   widget build(BuildContext context) {
54
95     body: Stack(
96       children: [
97         // Widget de Google Maps
98         GoogleMap(
99           onMapCreated: _onMapCreated,
100           initialCameraPosition: CameraPosition(
101             target: _center,
102             zoom: 14.0, // Nivel de zoom inicial
103           ), // CameraPosition
104           markers: _markers,
105           myLocationButtonEnabled: false, // Oculta el botón de mi ubicación
106           zoomControlsEnabled: false, // Oculta los controles de zoom por defecto
107         ), // GoogleMap
108
109         // Barra de búsqueda flotante
110         Positioned(
111           top: MediaQuery.of(context).padding.top + 70, // Ajusta la posición para que no se superponga con el AppBar
112           left: 16,
113           right: 16,
114           child: Container(
115             padding: const EdgeInsets.symmetric(horizontal: 16.0),
116             decoration: BoxDecoration(
117               color: Colors.white,
118               borderRadius: BorderRadius.circular(30),
119               boxShadow: [
120                 BoxShadow(
121                   color: Colors.black.withOpacity(0.1),
122                   blurRadius: 10,
123                   offset: const Offset(0, 5),
124                 ), // BoxShadow
125               ],
126             ), // BoxDecoration
127             child: TextField(
128               decoration: InputDecoration(

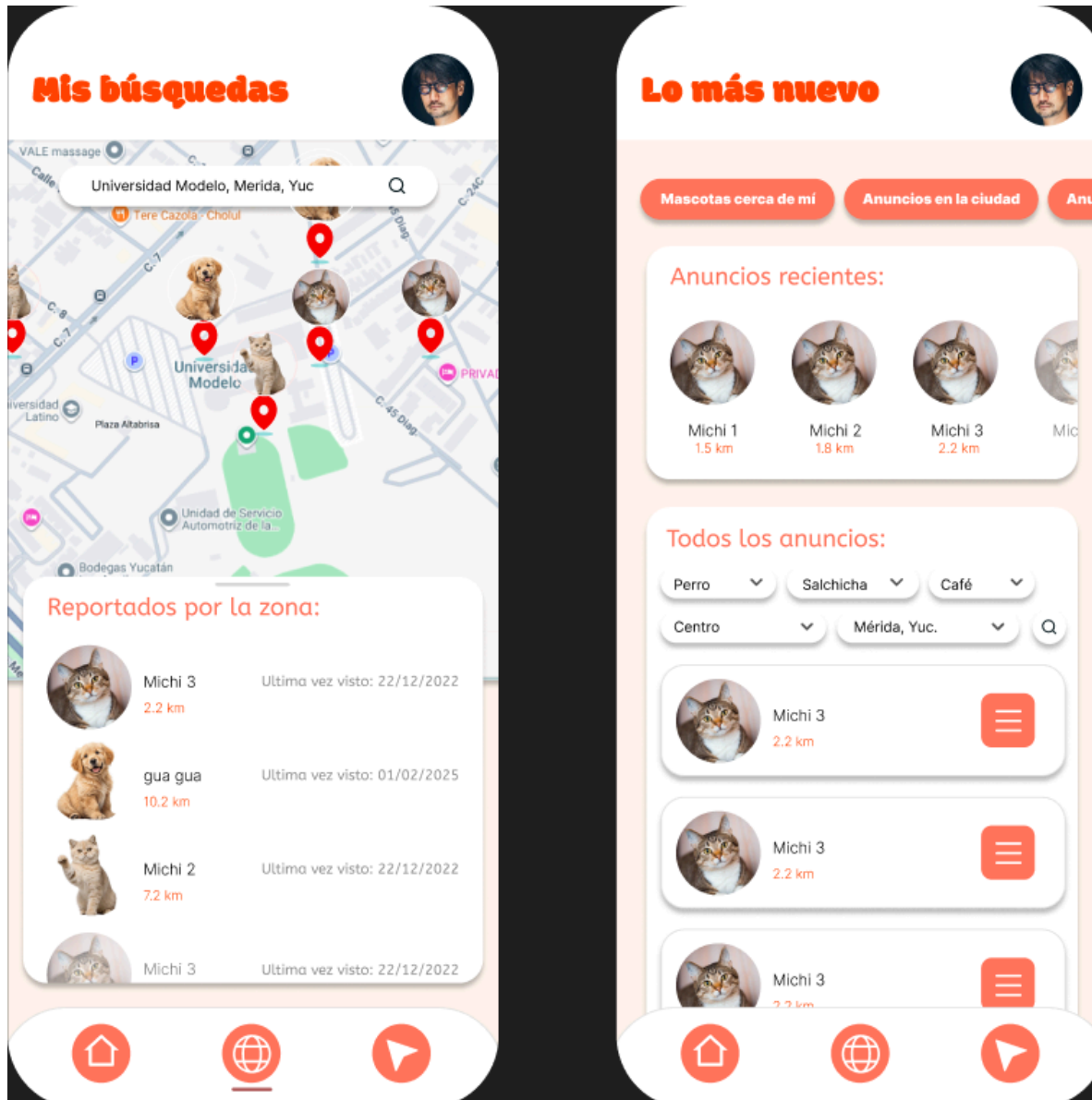
```

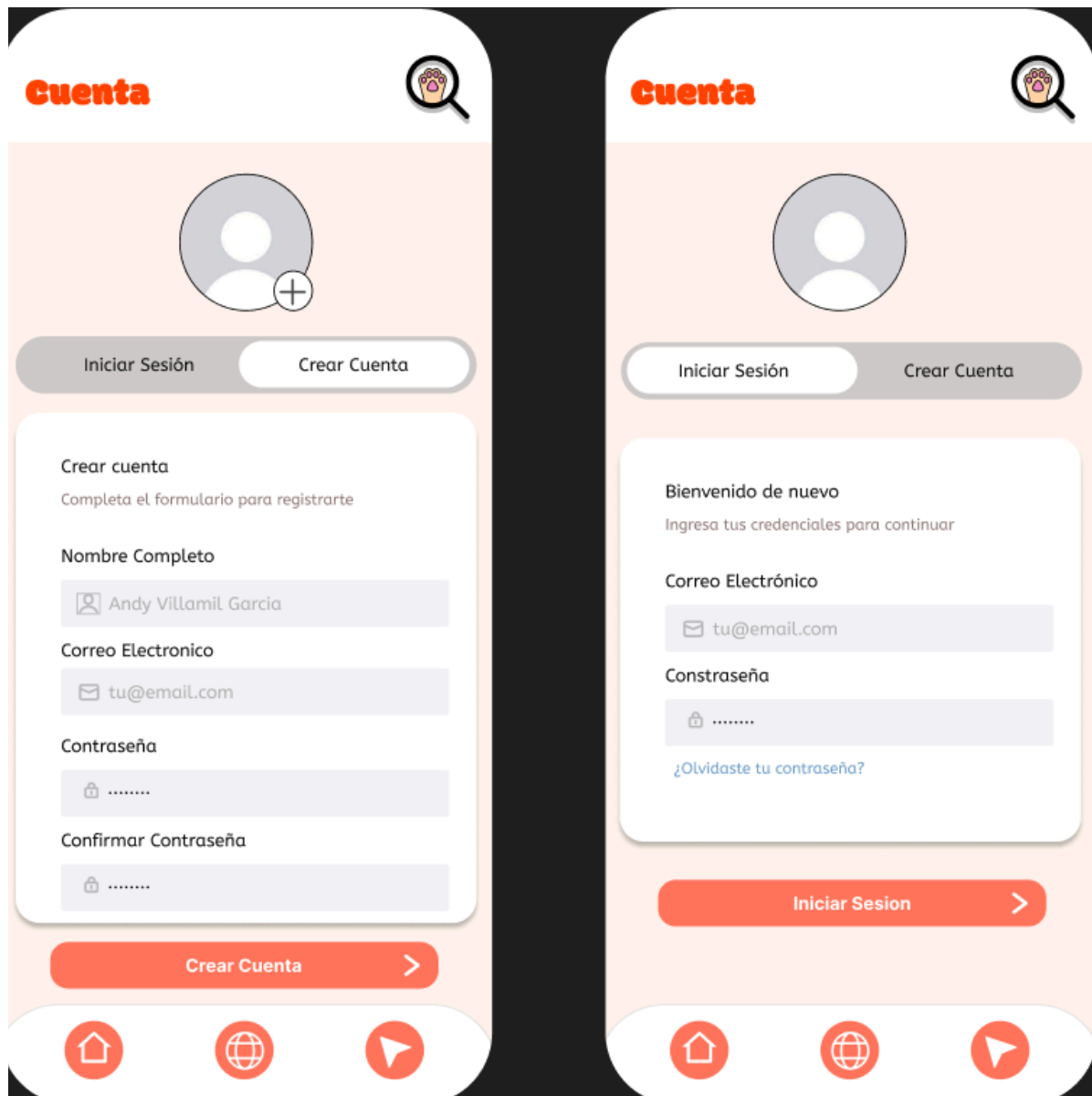
```
128 decoration: InputDecoration(
129   hintText: "Universidad Modelo, Merida, Yuc",
130   hintStyle: const TextStyle(color: kDarkGreyColor),
131   icon: Icon(Icons.search, color: kDarkGreyColor),
132   suffixIcon: Icon(Icons.more_vert, color: kDarkGreyColor), // ícono de tres puntos verticales
133   border: InputBorder.none,
134 ), // InputDecoration
135 ), // TextField
136 ), // Container
137 ), // Positioned
138 ],
139 ), // Stack
140 bottomNavigationBar: _buildBottomNavBar(context), // Pasamos el contexto
141 ); // Scaffold
142 }
```

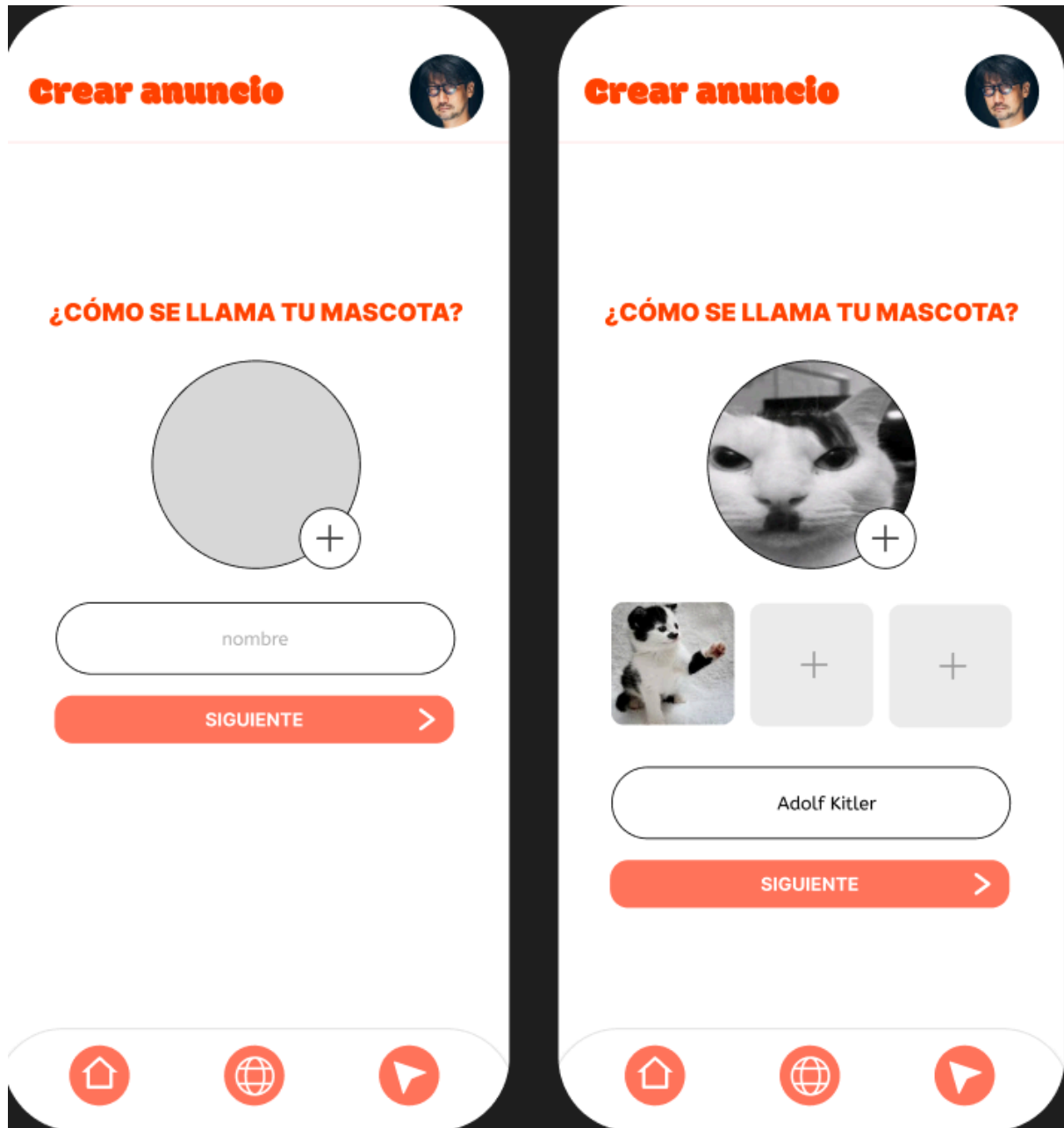
```
144 /// Widget para la Barra de Navegación Inferior
145 widget _buildBottomNavBar(BuildContext context) {
146   return Container(
147     decoration: BoxDecoration(
148       color: Colors.white,
149       borderRadius: const BorderRadius.vertical(top: Radius.circular(25)),
150       boxShadow: [
151         BoxShadow(
152           color: Colors.black.withOpacity(0.05),
153           blurRadius: 10,
154         ) // BoxShadow
155       ],
156     ), // BoxDecoration
157     clipBehavior: Clip.antiAlias,
158     child: BottomNavigationBar(
159       items: const <BottomNavigationBarItem>[
160         BottomNavigationBarItem(
161           icon: Icon(Icons.home_filled),
162           label: 'Home',
163         ), // BottomNavigationBarItem
164         BottomNavigationBarItem(
165           icon: Icon(Icons.public),
166           label: 'Discover',
167         ), // BottomNavigationBarItem
168         BottomNavigationBarItem(
169           icon: Icon(Icons.send_rounded),
170           label: 'Send',
171         ), // BottomNavigationBarItem
172       ], // <BottomNavigationBarItem>[]
173       currentIndex: 1, // Ahora el ícono del planeta (índice 1) estará seleccionado
174       selectedItemColor: kPrimaryColor,
175       unselectedItemColor: kDarkGreyColor,
176       backgroundColor: Colors.white,
177       showSelectedLabels: false,
178       showUnselectedLabels: false,
```

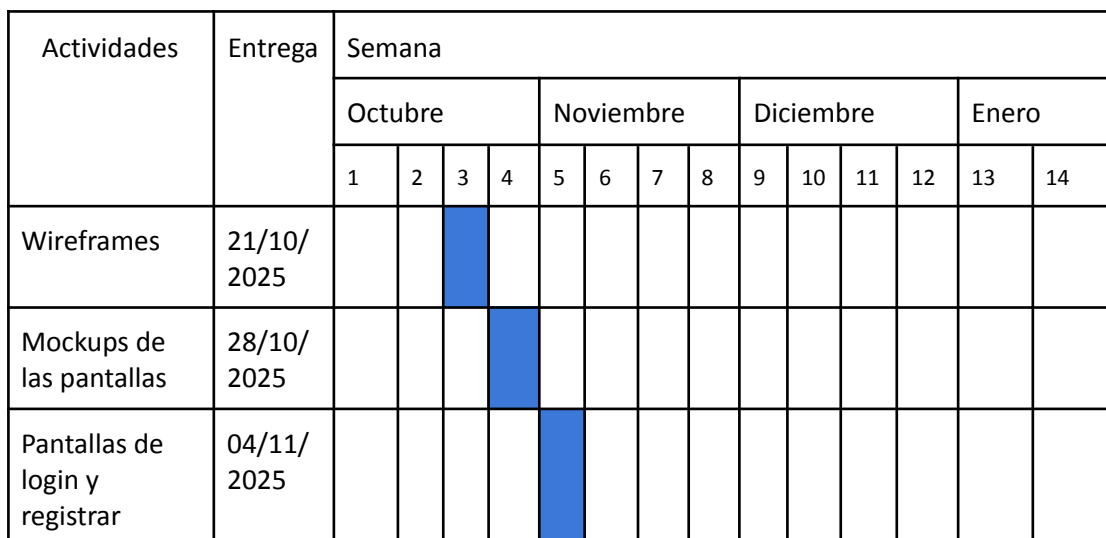
```
179       type: BottomNavigationBarType.fixed,
180       elevation: 0,
181       onTap: (index) {
182         if (index == 0) {
183           // Si el usuario toca "Home", volvemos a la pantalla de autenticación
184           Navigator.of(context).pushReplacement(
185             MaterialPageRoute(builder: (context) => AuthScreen()),
186           );
187         }
188         // Para el ícono del planeta (índice 1), no hacemos nada porque ya estamos aquí
189         // Para el ícono de "Send" (índice 2), puedes agregar otra navegación si es necesario
190       },
191     ), // BottomNavigationBar
192   ); // Container
193 }
194 }
```











[illegible]