

CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS – UNIMINUTO

Valentina Sánchez Rivera

Andrés Felipe Molano Montejo

Jetli Julian Mosquera

INGENIERIA DE SOFTWARE - PAWBOND

Bogotá, 28 de noviembre de 2025



Introducción

El presente proyecto propone el desarrollo de PawBOND, un collar inteligente diseñado para la identificación y el cuidado integral de mascotas mediante el uso de tecnologías como el chip NFC y una plataforma digital en la nube. Este producto busca ofrecer a los dueños una herramienta práctica y confiable que centralice información médica, datos de contacto y servicios complementarios para el bienestar animal. En el ámbito académico, este trabajo se enmarca como una iniciativa de innovación tecnológica aplicada a la gestión de información y el diseño de soluciones digitales con impacto social. Desde una perspectiva profesional, PawBOND se orienta a responder a la creciente demanda del mercado de productos inteligentes para mascotas, aportando un valor agregado a través de la seguridad, la conectividad y la interacción comunitaria entre propietarios y entidades relacionadas con el cuidado animal.

Planteamiento del problema

En la actualidad, miles de mascotas se extravían cada año y muchas no logran regresar a sus hogares debido a la falta de mecanismos de identificación eficientes. Los métodos tradicionales, como las placas metálicas o collares con grabados, presentan limitaciones: se deterioran con facilidad, no almacenan información relevante y dependen de la disponibilidad de la persona que encuentra al animal para hacer contacto con el dueño. Adicionalmente, la información médica de las mascotas suele estar dispersa en documentos físicos o en bases de datos aisladas de clínicas veterinarias, lo que dificulta un acceso rápido en casos de emergencia. Esto genera una necesidad creciente de soluciones tecnológicas que permitan centralizar datos, garantizar la identificación inmediata y mejorar la seguridad y el bienestar animal.

Objetivos

Objetivo general.

Desarrollar un sistema integral de identificación y cuidado de mascotas mediante un collar inteligente con chip NFC y una aplicación digital en la nube, que permita garantizar la seguridad, salud y bienestar de los animales, además de fortalecer la conexión entre dueños a través de una comunidad colaborativa.

Objetivos específicos

1. Diseñar una plataforma digital en la nube para la gestión de información médica de la mascota y los datos de contacto del dueño.
2. Incorporar un sistema de alertas y notificaciones personalizadas para mejorar la prevención, el cuidado y la localización de mascotas en caso de pérdida.
3. Garantizar la seguridad y protección de los datos de las mascotas y sus dueños, reduciendo riesgos de pérdida de información o mal uso de esta.
4. Fomentar la conectividad social entre los dueños de mascotas mediante una comunidad en línea que facilite el intercambio de experiencias y recomendaciones.

Diagramas de Comportamiento

Diagrama de Casos de Uso

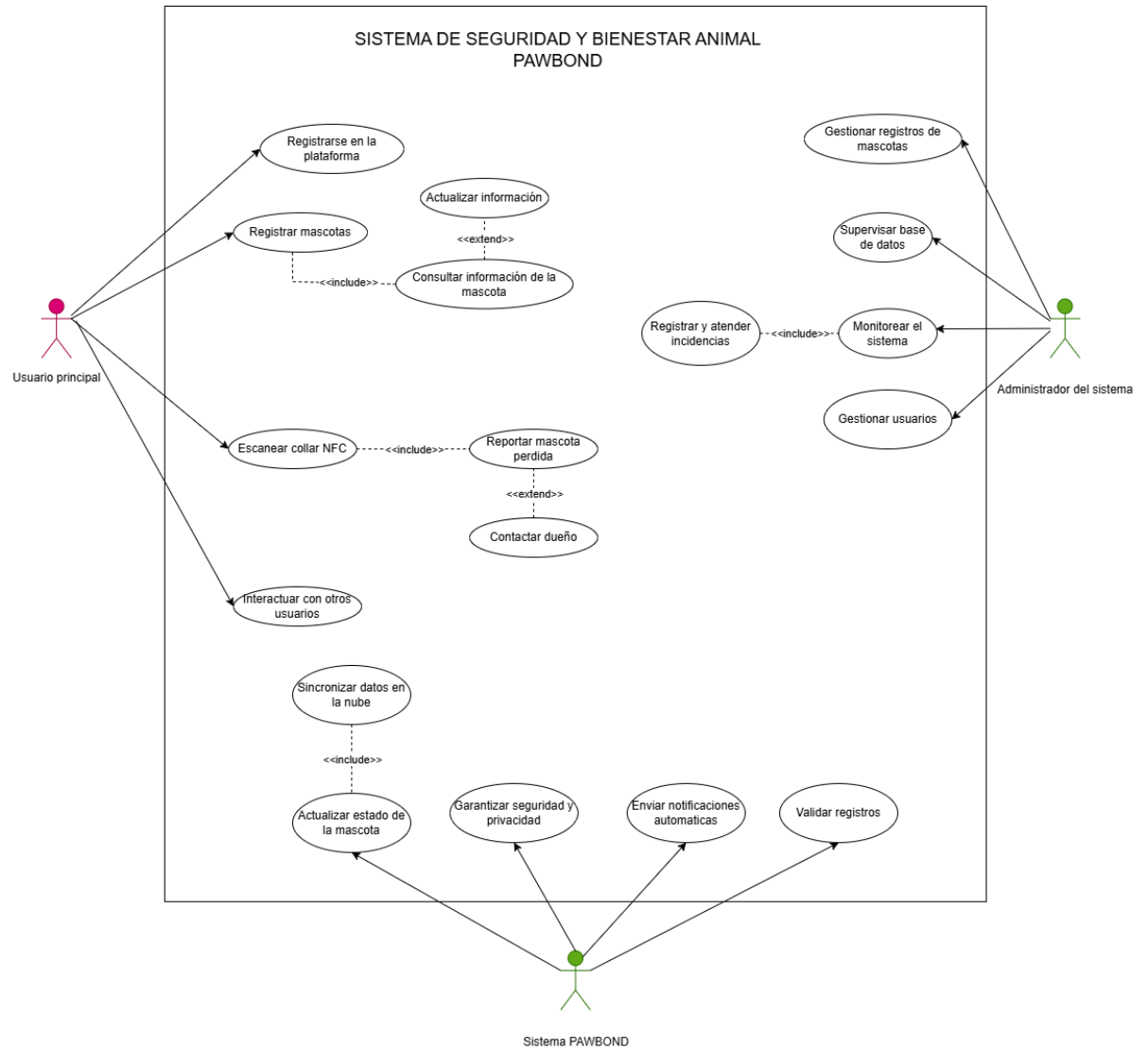
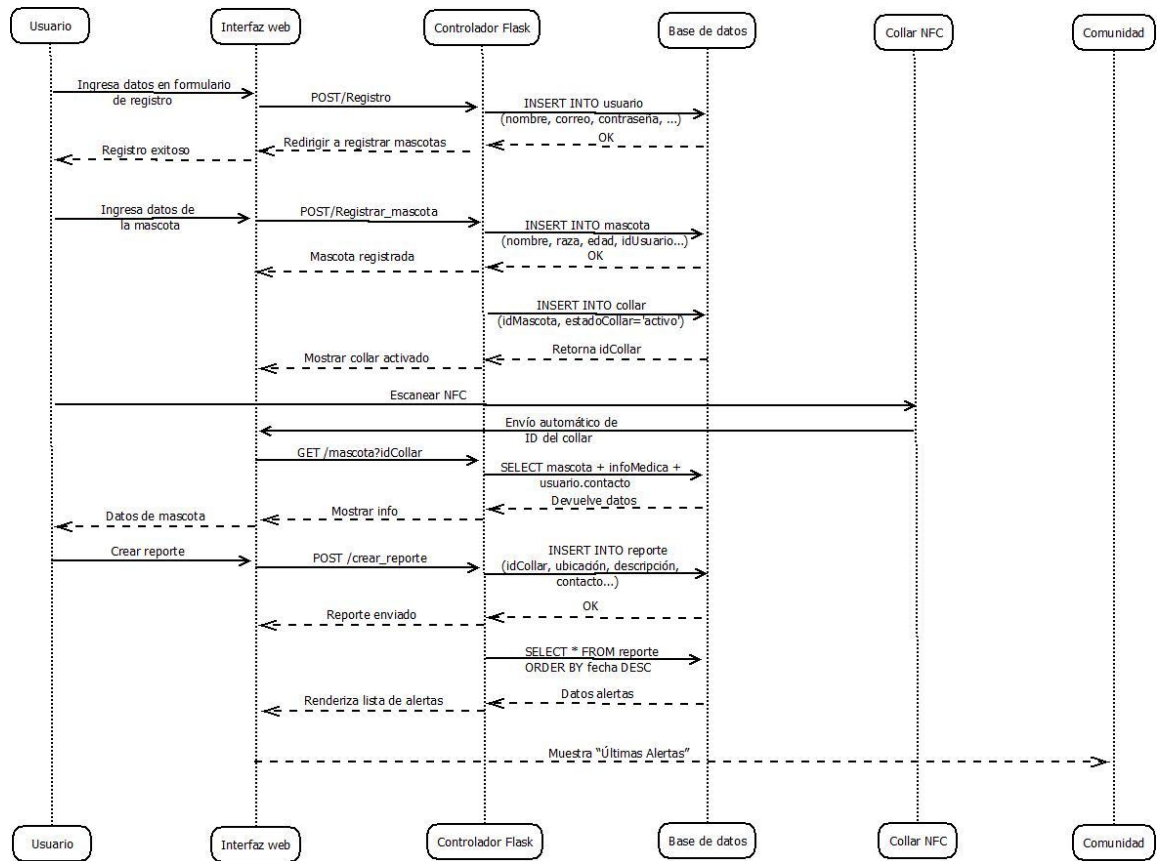


Diagrama de Secuencial



Diseño Estructural

Diagrama Modelo – Entidad

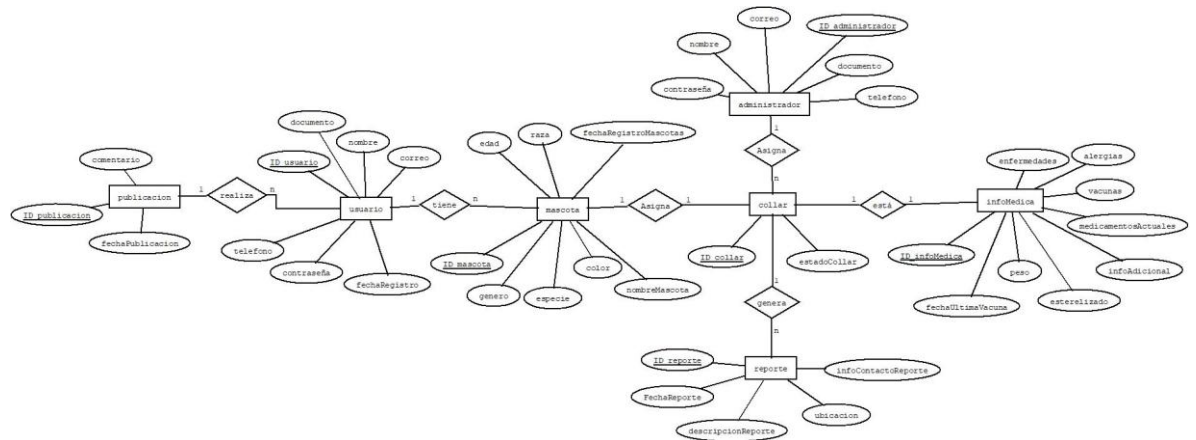
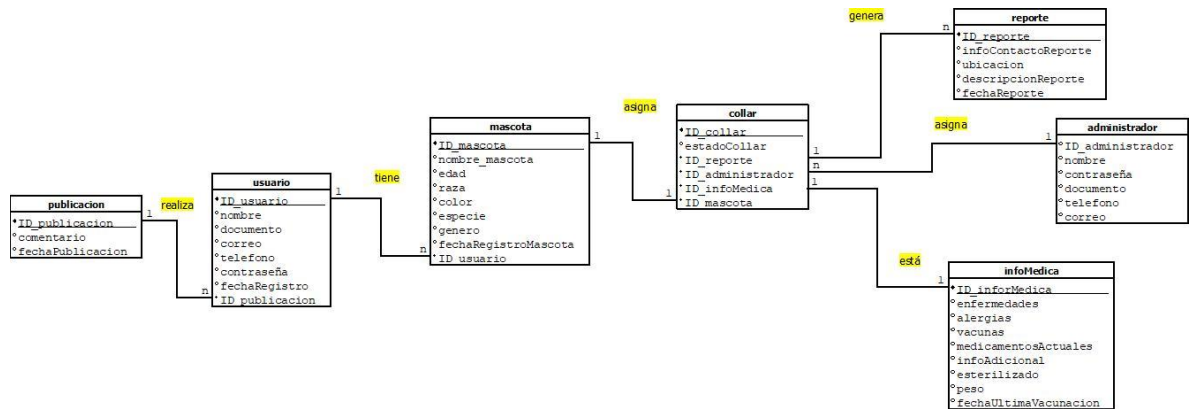
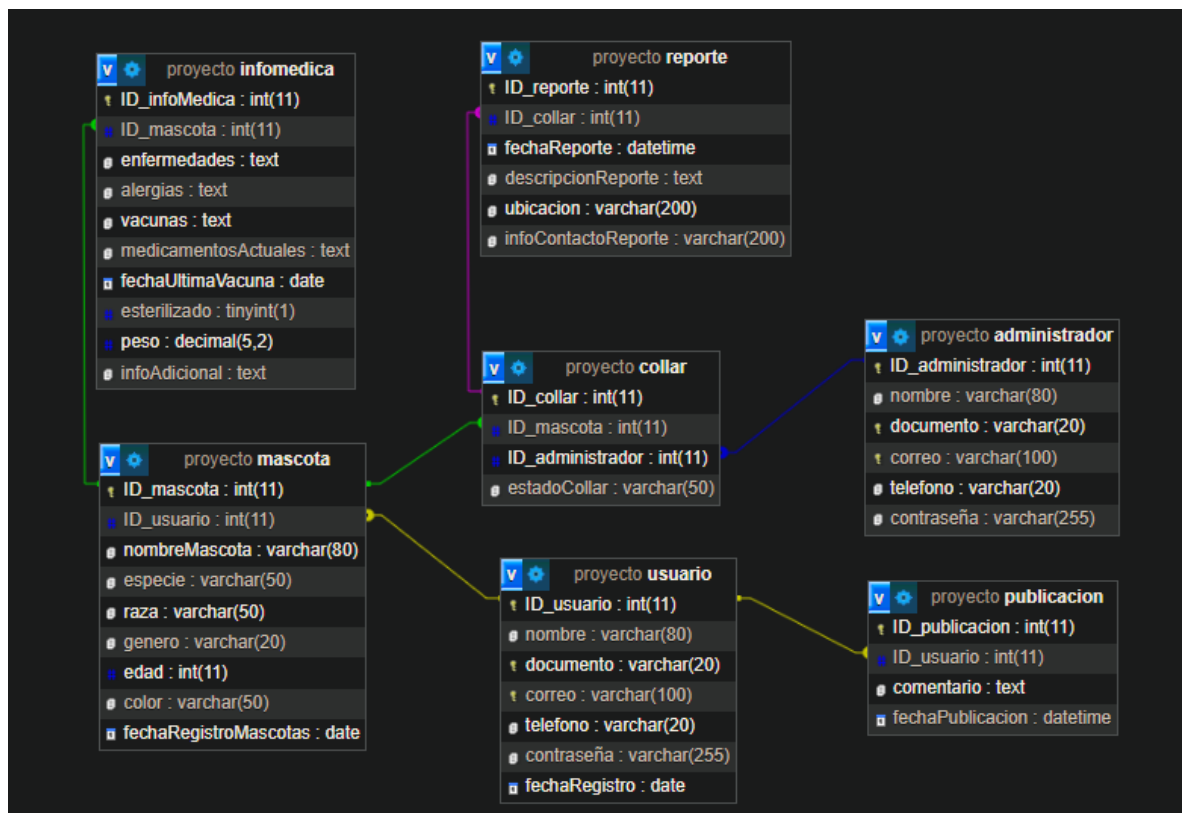


Diagrama Relacional



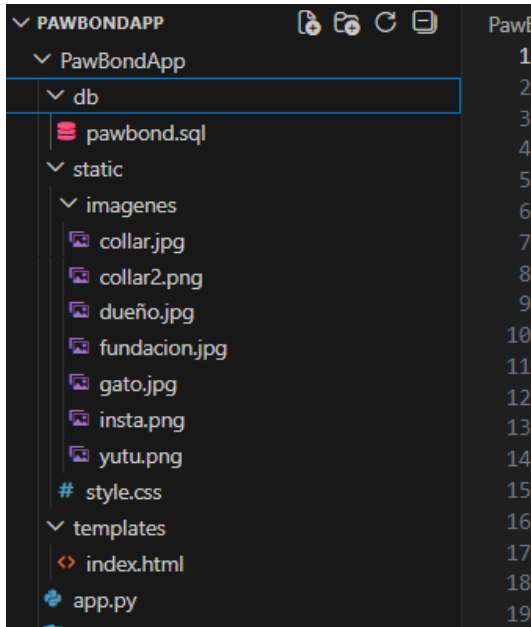
Modelo Físico



Prototipo Técnico (CRUD Flask + MySQL)

Estructura del proyecto Flask

1. Crear la estructura de carpetas



2. Crear base de datos en MySQL e insertar datos

```
1 -- CREAR TABLA USUARIOS
2 CREATE TABLE `usuarios` (
3   `idUsuario` INT AUTO_INCREMENT PRIMARY KEY,
4   `nombre` VARCHAR(100) NOT NULL,
5   `correo` VARCHAR(120) UNIQUE,
6   `contraseña` VARCHAR(255) NOT NULL,
7   `telefono` VARCHAR(20) DEFAULT NULL,
8   `numeroDocumento` VARCHAR(20) UNIQUE,
9   `fechaRegistro` DATETIME);
10

1 -- ADICIONAR CAMPO DIRECCION A LA TABLA USUARIO
2 ALTER TABLE usuario add COLUMN direccion VARCHAR(50) NOT NULL;

1 ALTER TABLE usuario
2 MODIFY COLUMN fechaRegistro DATETIME DEFAULT CURRENT_TIMESTAMP;
```

```
1 -- CREAR TABLA MASCOTA
2 CREATE TABLE `mascota` (
3   `idMascota` INT AUTO_INCREMENT PRIMARY KEY,
4   `nombreMascota` VARCHAR(100) NOT NULL,
5   `raza` VARCHAR(50) NOT NULL,
6   `edad` INT NOT NULL,
7   `genero` ENUM('Macho','Hembra'),
8   `especie` VARCHAR(50) NOT NULL,
9   `color` VARCHAR(50) NOT NULL,
10  `fechaRegistroMascota` DATETIME DEFAULT NULL,
11  `idUsuario` INT);
12
```

```
1 ALTER TABLE mascota
2 MODIFY COLUMN fechaRegistroMascota DATETIME DEFAULT CURRENT_TIMESTAMP;
```

```
1 -- CREAR TABLA COLLAR
2 CREATE TABLE `collar` (
3   `idCollar` INT AUTO_INCREMENT PRIMARY KEY,
4   `idMascota` INT NOT NULL,
5   `codigoNFC` INT NOT NULL,
6   `estadoCollar` ENUM('activo','inactivo'));
7
```

```
1 -- CREAR TABLA INFOMEDICA
2 CREATE TABLE `InfoMedica` (
3   `idInfo` INT AUTO_INCREMENT PRIMARY KEY,
4   `idMascota` INT,
5   `peso` TEXT(100) NULL,
6   `medicamentosActuales` TEXT(100) NULL,
7   `alergias` TEXT(100) NULL,
8   `enfermedades` TEXT(100) NULL,
9   `vacunas` TEXT(100) NULL,
10  `fechaUltimaVacuna` DATE DEFAULT NULL,
11  `esterilizado` ENUM ('Si', 'No'),
12  `infoAdicional` TEXT(100) NULL );
```



```

1 CREATE TABLE administrador (
2     id_administrador INT NOT NULL AUTO_INCREMENT,
3     nombre VARCHAR(100) NOT NULL,
4     contrasena VARCHAR(255) NOT NULL,
5     correo VARCHAR(100) UNIQUE NOT NULL,
6     documento VARCHAR(20) UNIQUE NOT NULL,
7     telefono VARCHAR(15),
8     PRIMARY KEY (id_administrador)
9 );|

```

```

1 -- CREAR TABLA PUBLICACION|
2 CREATE TABLE `publicacion` (
3     `idPublicacion` INT AUTO_INCREMENT PRIMARY KEY,
4     `idUsuario` INT,
5     `Comentario` TEXT(120) NOT NULL,
6     `FechaPublicacion` DATETIME DEFAULT NULL);

```

```

1 ALTER TABLE publicacion
2 MODIFY COLUMN FechaPublicacion DATETIME DEFAULT CURRENT_TIMESTAMP;

```

```

1 -- CREAR TABLA REPORTE
2 CREATE TABLE `reporte` (
3     `idReporte` INT AUTO_INCREMENT PRIMARY KEY,
4     `idCollar` INT,
5     `ubicacion` VARCHAR(50) NOT NULL,
6     `descripcionReporte` TEXT(120) NOT NULL,
7     `infoContactoReporte` VARCHAR(50) NOT NULL,
8     `fechaReporte` DATETIME DEFAULT NULL);

```

```

1 ALTER TABLE reporte
2 MODIFY COLUMN fechaReporte DATETIME DEFAULT CURRENT_TIMESTAMP;

```

LLAVES FORANEAS

```

1 -- LLAVE FORANEA DE USUARIO Y MASCOTA
2 ALTER TABLE mascota ADD
3 CONSTRAINT tiene
4 FOREIGN KEY (idUserio)
5 REFERENCES usuario (idUserio)

```

```

1 -- LLAVE FORANEA DE MASCOTA Y COLLAR
2 ALTER TABLE collar ADD
3 CONSTRAINT pertenece
4 FOREIGN KEY (idMascota)
5 REFERENCES mascota (idMascota)
6

```

3. Conectar Flask a la base de datos

```

PawBondApp > app.py > listar_mascotas
1 from flask import Flask, render_template, request, redirect, url_for, flash
2 from flask_mysql import MySQL
3 import datetime
4
5 app = Flask(__name__)
6 app.secret_key = 'mi_clave_secreta_super_segura'
7
8 # CONFIGURACIÓN DE LA BASE DE DATOS
9 app.config['MYSQL_HOST'] = 'localhost'
10 app.config['MYSQL_USER'] = 'root'
11 app.config['MYSQL_PASSWORD'] = ''
12 app.config['MYSQL_DB'] = 'pawbond'
13 app.config['MYSQL_CURSORCLASS'] = 'DictCursor'
14 mysql = MySQL(app)
15
16 @app.route('/')
17 def listar_mascotas():
18     mascotas = []
19     publicaciones = []
20
21     try:
22         cur = mysql.connection.cursor()
23
24         query_mascotas = """
25         SELECT
26             m.nombreMascota,
27             m.especie,
28             m.raza,
29             m.edad,
30             m.genero,
31             m.color,
32             u.nombre AS nombreDueno,
33             u.telefono AS telefonoDueno,
34             c.estadoCollar
35         FROM
36             mascota m
37         JOIN
38             usuario u ON m.idUsuario = u.idUsuario
39         LEFT JOIN
40             collar c ON m.idMascota = c.idMascota;
41         """
42         cur.execute(query_mascotas)
43         mascotas = cur.fetchall()
44
45         query_publicaciones = """

```

4. Desarrollo y funcionamiento del CRUD

- Crear (Create)

```
index.html X essay
PawBondApp > templates > index.html > @html
2 <html lang="es" style="width: 1000px; margin: 0 auto;">
154
155 <!-- Aquí dejamos la parte que va conectada a la db de los usuarios, así los registramos y quedan en la base, necesario para hacer acciones después -->
156
157 <!-- registro de usuario -->
158
159
160 <section id="registro_usuario" style="font-family:Helvetica (0); background-color: #fff07c7; padding: 30px 0;">
161 <div class="titulo_borde"><h3 class="titulos" style="text-align:center;"> Registrate Aquí ( Dunfo)/</h3></div>
162
163 <div class="form_registro_user" style="width: 500px; margin: 20px auto; padding: 20px; border: 1px solid #808075fc; border-radius: 8px; background-color: #c0c0e7e7;">
164 <form method="POST" action="{url_for('registrar_usuario')}">
165
166 <label for="nombre">Nombre Completo:</label>
167 <input type="text" id="nombre" name="nombre" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
168
169 <label for="correo">Correo Electrónico:</label>
170 <input type="email" id="correo" name="correo" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
171
172 <label for="clave">Contraseña:</label>
173 <input type="password" id="clave" name="clave" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
174
175 <label for="documento">Número de Documento:</label>
176 <input type="text" id="documento" name="documento" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
177
178 <label for="telefono">Teléfono:</label>
179 <input type="number" id="telefono" name="telefono" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
180
181 <label for="direccion">Dirección:</label>
182 <input type="text" id="direccion" name="direccion" required style="width: 100%; padding: 8px; margin-bottom: 20px; box-sizing: border-box;">
183
184 <button type="submit" style="background-color: #f3f3f3; color: #000; padding: 10px 15px; border: none; border-radius: 5px; cursor: pointer; width: 100%;">Registrar Usuario</button>
185
186 </div>
187
188 </div>
189 </div>
190 </section>
191
```

```
index.html app.py X
PawBondApp > app.py > listar_mascotas
107
108 @app.route('/registrar_usuario', methods=['POST'])
109 def registrar_usuario():
110     if request.method == 'POST':
111         try:
112             cur = mysql.connection.cursor()
113
114             nombre_u = request.form['nombre']
115             correo_u = request.form['correo']
116             clave_u = request.form['clave']
117             doc_u = request.form['documento']
118             tel_u = request.form['telefono']
119             dir_u = request.form['direccion']
120
121
122             query_usuario = """
123             INSERT INTO usuario (nombre, correo, contraseña, numeroDocumento, telefono, direccion, fechaRegistro)
124             VALUES (%s, %s, %s, %s, %s, %s, %s)
125             """
126             fecha_registro = datetime.datetime.now()
127             cur.execute(query_usuario, (nombre_u, correo_u, clave_u, doc_u, tel_u, dir_u, fecha_registro))
128
129             mysql.connection.commit()
130             cur.close()
131
132             flash(f'Registro de usuario exitoso! Bienvenido/a {nombre_u}.', 'success')
133
134         except Exception as e:
135             mysql.connection.rollback()
136             print(f'Error durante el registro de usuario: {e}')
137             flash(f'Error al registrar el usuario: {e}', 'error')
138
139     return redirect(url_for('listar_mascotas'))
140
```

```

142 @app.route('/registrar_mascota', methods=['POST'])
143 def registrar_mascota():
144     if request.method == 'POST':
145         nombre_u = request.form['nombreDueno']
146
147         nombre_m = request.form['nombreMascota']
148         raza_m = request.form['raza']
149         edad_m = request.form['edad']
150         genero_m = request.form['genero']
151         especie_m = request.form['especie']
152         color_m = request.form['color']
153         esterilizado_m = request.form.get('esterilizado', 'No')
154
155     try:
156         cur = mysql.connection.cursor()
157
158         cur.execute("SELECT idUsuario FROM usuario WHERE nombre = %s", (nombre_u,))
159         usuario_data = cur.fetchone()
160
161         if not usuario_data:
162             flash(f'Error: El dueño "{nombre_u}" no está registrado. Regístrate primero.', 'error')
163             cur.close()
164             return redirect(url_for('listar_mascotas'))
165
166         id_usuario = usuario_data['idUsuario']
167
168         query_mascota = """
169         INSERT INTO mascota (idUsuario, nombreMascota, raza, edad, genero, especie, color)
170         VALUES (%s, %s, %s, %s, %s, %s, %s)
171         """
172         cur.execute(query_mascota, (id_usuario, nombre_m, raza_m, edad_m, genero_m, especie_m, color_m))
173
174         mysql.connection.commit()
175         cur.close()
176
177         flash(f'Mascota {nombre_m} registrada exitosamente!', 'success')
178
179     except Exception as e:
180         mysql.connection.rollback()
181         print(f"Error al registrar mascota: {e}")
182         flash('Error interno al registrar la mascota.', 'error')
183
184     return redirect(url_for('listar_mascotas'))
185

```

```

index.html X  app.py
PawBordApp > templates > @ index.html > @ html
1  <html lang="es" style="width: 1000px; margin: 0 auto;">
...
192 <!-- registro de mascota -->
193
194
195 <section id="registro_mascota" style="font-family:Helvetica; background-color: #f0f0f0; padding: 30px 0;><br>
196 <div class="titulo"><h2 class="titulo" style="text-align:center;">% Registra tu Mascota</h2></div>
197
198 <div class="form_registro_mascota" style="width: 500px; margin: 20px auto; padding: 20px; border: 1px solid #ccc; border-radius: 8px; background-color: #ffffff;">
199 <form method="POST" action="{{ url_for('registrar_mascota') }}">
200
201 <h3 style="color: #333333;">Datos del Dueño y Mascota</h3>
202
203 <label for="nombreDueno">Tu Nombre (Debe estar registrado)</label>
204 <input type="text" id="nombreDueno" name="nombreDueno" required style="width: 100%; padding: 8px; margin-bottom: 20px; box-sizing: border-box;">
205
206 <label for="nombreMascota">Nombre de la Mascota</label>
207 <input type="text" id="nombreMascota" name="nombreMascota" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
208
209 <label for="especie">Especie</label>
210 <select id="especie" name="especie" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
211 <option value="Perro">Perro</option>
212 <option value="Gato">Gato</option>
213 <option value="Otro">Otro</option>
214 </select>
215
216 <label for="raza">Raza</label>
217 <input type="text" id="raza" name="raza" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
218
219 <label for="edad">Edad (Años)</label>
220 <input type="number" id="edad" name="edad" min="0" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
221
222 <label for="genero">Género</label>
223 <select id="genero" name="genero" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
224 <option value="Macho">Macho</option>
225 <option value="Hembra">Hembra</option>
226 </select>
227
228 <label for="color">Color</label>
229 <input type="text" id="color" name="color" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
230
231 <label for="esterilizado">¿Está esterilizado? (Validación simple)</label>
232 <select id="esterilizado" name="esterilizado" style="width: 100%; padding: 8px; margin-bottom: 20px; box-sizing: border-box;">
233 <option value="No">No</option>
234 <option value="Sí">Sí</option>
235 </select>
236
237 <button type="submit" style="background-color: #333333; color: white; padding: 10px 15px; border: none; border-radius: 5px; cursor: pointer; width: 100%;">Registrar Mascota</button>
238 </form>
239 </div>
240 <br>
241 <br>
242 </section>
243

```

- (READ) (UPDATE) (DELETE) En nuestra web de las acciones de leer/actualizar y borrar usuario “(READ) (UPDATE) (DELETE)” están evidenciados en el siguiente apartado.

```

69 @app.route('/publicar', methods=['POST'])
70 def publicar_alerta():
71     if request.method == 'POST':
72         nombre_u = request.form['nombreUsuario']
73         comentario = request.form['comentario']
74
75     try:
76         cur = mysql.connection.cursor()
77
78         cur.execute("SELECT idUsuario FROM usuario WHERE nombre = %s", (nombre_u,))
79         usuario_data = cur.fetchone()
80
81         if not usuario_data:
82             flash(f'Error: El usuario "{nombre_u}" no está registrado.', 'error')
83             cur.close()
84             return redirect(url_for('listar_mascotas'))
85
86         id_usuario = usuario_data['idUsuario']
87
88         query_publicacion = """
89         INSERT INTO publicacion (idUsuario, Comentario)
90         VALUES (%s, %s)
91         """
92         cur.execute(query_publicacion, (id_usuario, comentario))
93
94
95         mysql.connection.commit()
96         cur.close()
97
98         flash('¡Alerta publicada exitosamente!', 'success')
99
100     except Exception as e:
101         mysql.connection.rollback()
102         print(f"Error al publicar alerta: {e}")
103         flash('Error interno al publicar la alerta.', 'error')
104
105     return redirect(url_for('listar_mascotas'))

```

```

16 @app.route('/')
17 def listar_mascotas():
18     mascotas = []
19     publicaciones = []
20
21     try:
22         cur = mysql.connection.cursor()
23
24         query_mascotas = """
25         SELECT
26             m.nombreMascota,
27             m.especie,
28             m.raza,
29             m.edad,
30             m.genero,
31             m.color,
32             u.nombre AS nombreDueño,
33             u.telefono AS telefonoDueño,
34             c.estadoCollar
35         FROM
36             mascota m
37         JOIN
38             usuario u ON m.idUsuario = u.idUsuario
39         LEFT JOIN
40             collar c ON m.idMascota = c.idMascota;
41         """
42         cur.execute(query_mascotas)
43         mascotas = cur.fetchall()
44
45         query_publicaciones = """
46         SELECT
47             p.Comentario,
48             p.FechaPublicacion,
49             u.nombre AS nombreUsuario
50         FROM
51             publicacion p
52         JOIN
53             usuario u ON p.idUsuario = u.idUsuario
54         ORDER BY
55             p.FechaPublicacion DESC;
56         """
57         cur.execute(query_publicaciones)
58         publicaciones = cur.fetchall()
59
60         cur.close()
61
62     except Exception as e:
63         print(f"Error al listar datos: {e}")
64         flash("Error al cargar datos de la base de datos.", 'error')
65
66     return render_template('index.html', mascotas=mascotas, publicaciones=publicaciones)

```

```

245 <!-- Aquí dejamos la parte que va conectada a la db donde estan las publicaciones -->
246
247 <section id="crear_publicacion" style="font-family:Malandra GD; background-color: #f0f0f0; border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;">
248     <div class="titulosborde"><h2 class="Titulos" style="text-align:center;">Crea una Alerta </h2></div>
249     <p style="text-align: center;">Usa este formulario para reportar un avistamiento o una pérdida de mascota.</p>
250
251     <div class="form_publicacion" style="width: 50%; margin: 0 auto; padding: 20px; border: 1px solid #ccc; border-radius: 8px; background-color: #fff;">
252         <form method="POST" action="{url_for('publicar_alerta')}">
253             <label for="nombreUsuario">Tu Nombre (Debe estar registrado)</label><br>
254             <input type="text" id="nombreUsuario" name="nombreUsuario" required
255                 style="width: 100%; padding: 8px; margin-bottom: 15px; border: 1px solid #ccc;">
256
257             <label for="comentario">Mensaje / Alerta (Máx. 250 caracteres)</label><br>
258             <textarea id="comentario" name="comentario" rows="4" maxlength="250" required
259                 style="width: 100%; padding: 8px; margin-bottom: 20px; border: 1px solid #ccc;">
260
261             <button type="submit"
262                 style="background-color: #360101; color: white; padding: 10px 15px; border: none; border-radius: 5px; cursor: pointer;">
263                 Publicar Alerta
264             </button>
265         </form>
266     </div>
267     <br>
268     <hr>
269 </section>
270
271
272
273 <section id="novedades" style="font-family:Malandra GD; background-color: #f0f0f0; padding: 10px;">
274     <div class="titulosborde"><h2 class="Titulos" style="text-align:center;">Últimas Alertas de la Comunidad</h2></div>
275
276     <div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;">
277         <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px;">
278             <div style="display: flex; justify-content: space-between; align-items: center;">
279                 <div style="flex: 1;">
280                     <p style="margin: 0; font-size: 1.1em;">{{ publicacion.Comentario }}
281                 </div>
282                 <div style="flex: 0 0 150px; text-align: right; font-size: 0.9em; color: #ccc;">
283                     Publicado por <b>{{ publicacion.nombreUsuario }}</b> el <b>{{ publicacion.FechaPublicacion }}</b>
284                 </div>
285             </div>
286         </div>
287         <div style="padding-top: 5px;">
288             <div style="display: flex; justify-content: space-between; align-items: center;">
289                 <div style="flex: 1;">
290                     <p style="margin: 0; font-size: 1.1em;">{{ publicacion.Comentario }}
291                 </div>
292                 <div style="flex: 0 0 150px; text-align: right; font-size: 0.9em; color: #ccc;">
293                     Publicado por <b>{{ publicacion.nombreUsuario }}</b> el <b>{{ publicacion.FechaPublicacion }}</b>
294                 </div>
295             </div>
296         </div>

```

Explicación de la conexión con MySQL

Por medio del documento app.py tenemos, por CRUD-Flask, una conexión a la base de datos que validamos mediante XAMPP. Esta conexión hace posible guardar registros en tiempo real de mascotas y usuarios, así como generar publicaciones que actualizan la pantalla principal de la página, y todos estos datos los podemos gestionar desde la base de datos.

Descripción de la interfaz base

El documento HTML principal index no es solo una página estática, sino una interfaz dinámica que actúa como el puente de comunicación entre el usuario y la plataforma de gestión del proyecto. En su diseño, la página cumple una doble función clave:

- **Presentación y Justificación del Proyecto**

La primera mitad de la página se dedica a vender la idea y a informar al visitante. Propuesta de Valor: Se describe el producto central: un collar inteligente con chip NFC que garantiza la seguridad y el bienestar de las mascotas al permitir el acceso rápido a su información vital y los datos de contacto de sus dueños en caso de pérdida. Segmento de Clientes: Se delimita claramente a quién va dirigida la solución (dueños preocupados y organizaciones/fundaciones), explicando cómo el proyecto se relaciona con ellos. Modelo de Negocio: Se detallan las fuentes de ingreso (venta del collar, publicidad y suscripciones premium) y se incluye un formulario de Validación para medir el interés del público.

- **Interacción Directa con la Comunidad y la Base de Datos**

La segunda mitad de la página transforma la landing page en una herramienta de gestión activa que se conecta directamente con el sistema de PawBond. Registro de la Comunidad: Proporciona formularios sencillos para que los dueños puedan crear sus perfiles y, posteriormente, registrar a sus mascotas con datos esenciales (raza, edad, estado de esterilización, etc.). Esto sienta las bases para construir una comunidad de usuarios responsables. Sistema de Alertas: Ofrece un formulario para Crear una Alerta (reportar una pérdida o un avistamiento). Lo más importante es la sección "Últimas Alertas de la Comunidad" (#novedades), que funciona como un muro de publicaciones en tiempo real. Muestra las alertas enviadas por otros usuarios, permitiendo que la información fluya instantáneamente y la comunidad pueda actuar de forma coordinada.

Evidencias integración

Evidencia de ejecución del CRUD

Prueba inicial del registro de alertas
(Así se haria un registro de usuario/mascota, ya estando registro podemos hacer alertas)

Regístrate Aquí (Dueño)

Nombre Completo:
Jose Maria Cordoba

Correo Electrónico:
joseito@gmail.com

Contraseña:

Número de Documento:
12548655789

Teléfono:
3002504794

Dirección:
cr79# 14 sur

Registrar Usuario

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

		idUsuario	nombre	correo	contraseña	telefono	numeroDocumento	fechaRegistro	direccion
<input type="checkbox"/>	Editar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		4	Laura Perez	laura@gmail.com	123456	2147483647	10255378954	2025-02-14 00:00:00	Calle 10 # 5-51
<input type="checkbox"/>	Editar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		5	Carltos Gutierrez	carlos@gmail.com	abc123	2147483647	1028695874	2025-11-14 00:00:00	Calle 10 # 5-51
<input type="checkbox"/>	Editar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		6	Luisito Comunica	luisito@gmail.com	abc123	2147483647	5228656950	2024-12-10 00:00:00	Carrera 21 # 17-63
<input checked="" type="checkbox"/>	Editar	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
		7	Jose Maria Cordoba	joseito@gmail.com	123456789	2147483647	12548655789	2025-11-25 21:40:04	cr79# 14 sur

Seleccionar todo | Para los elementos que están marcados: Editar Copiar Borrar Exportar

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Operaciones sobre los resultados de la consulta

¡Crea una alerta!

Usa este formulario para reportar un avistamiento o una pérdida de mascota.

Tu Nombre (Debe estar registrado):

Jose Maria Cordoba

Mensaje / Alerta (Máx. 250 caracteres):

ENTREGA FINAL PROGRAMACION WEB

Publicar Alerta

Últimas Alertas de la Comunidad

"Busco a mi Poodle Coco. Se perdió hace una hora, por favor escanear el chip si la ven."

Publicado por **Luisito Comunica** el 2025-11-15 17:45:00

"Encontré un gato Siames con collar NFC en la Calle 5. Contacté al dueño, estoy esperando respuesta."

Publicado por **Carlitos Gutierrez** el 2025-11-15 16:10:00

"¡Alerta! Mi perro Max fue visto cerca del parque central. Collar NFC activo."

Publicado por **Laura Perez** el 2025-11-15 15:30:00



Click si quieres saber mas

¡Crea una Alerta! ⚠

Usa este formulario para reportar un avistamiento o una pérdida de mascota.

Tu Nombre (Debe estar registrado):

Mensaje / Alerta (Máx. 250 caracteres):

Publicar Alerta

Últimas Alertas de la Comunidad

"ENTREGA FINAL PROGRMACION WEB"

Publicado por Jose María Cordoba el 2025-11-25 21:41:37

"Busco a mi Poodle Coco. Se perdió hace una hora, por favor escanear el chip si la ven."

Publicado por Luisito Comunica el 2025-11-15 17:45:00

"Encontré un gato Siamés con collar NFC en la Calle 5. Contacté al dueño, estoy esperando respuesta."

Publicado por Carlitos Gutierrez el 2025-11-15 16:10:00

"¡Alerta! Mi perro Max fue visto cerca del parque central. Collar NFC activo."

Publicado por Laura Perez el 2025-11-15 15:30:00

Nueva

collar

infomedica

mascota

publicacion

reporte

usuario

performance_schema

phpmyadmin

test

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

Mostrar todoNúmero de filas: 25

Filtrar filas:

Opciones extra

<div>←T→</div>	idPublicacion	idUsuario	Comentario	FechaPublicacion
<input type="checkbox"/> Editar Copiar Borrar	1	4	¡Alerta! Mi perro Max fue visto cerca del parque c...	2025-11-15 15:30:00
<input type="checkbox"/> Editar Copiar Borrar	2	5	Encontré un gato Siamés con collar NFC en la Calle...	2025-11-15 16:10:00
<input type="checkbox"/> Editar Copiar Borrar	3	6	Busco a mi Poodle Coco. Se perdió hace una hora, p...	2025-11-15 17:45:00
<input type="checkbox"/> Editar Copiar Borrar	4	7	ENTREGA FINAL PROGRMACION WEB	2025-11-25 21:41:37

☐ Seleccionar todo

Para los elementos que están marcados: [Editar](#) [Copiar](#) [Borrar](#) [Exportar](#)

Mostrar todoNúmero de filas: 25

Filtrar filas:

Desafíos y Aprendizajes

Durante el desarrollo de PawBOND enfrentamos desafíos técnicos, especialmente en la integración del collar NFC con la plataforma web y en el diseño de la base de datos robusta que soportara múltiples actores y procesos. Sin embargo, estos retos nos

permitieron fortalecer nuestras habilidades en modelado UML, optimización de bases de datos relacionales y desarrollo modular. También aprendimos a implementar buenas prácticas de arquitectura con Flask, gestionando rutas, controladores y seguridad de datos.

Conclusión

Creemos que PawBOND representa una solución digital altamente pertinente y con impacto social. El proyecto aborda la problemática de la identificación ineficiente y la dispersión de datos médicos de las mascotas, proponiendo una alternativa innovadora que integra un collar inteligente NFC con una plataforma en la nube. Nuestra elección del Modelo Incremental garantiza el desarrollo modular y la validación temprana con los usuarios, minimizando riesgos. En definitiva, nos enfocamos en diseñar un sistema que no solo garantice la seguridad y la localización inmediata de la mascota, sino que también fortalezca la comunidad de dueños, sentando las bases para un modelo de negocio sostenible en el cuidado animal. Además, el uso de diagramas en la planeación del software permitió visualizar, descomponer y comprender las interacciones del sistema antes de programar. La combinación del modelado UML con un desarrollo ágil en Flask permite que los equipo se comuniquen de una mejor manera, disminuye errores en etapas tempranas y permite que los cambios solicitados por los usuarios se integren sin comprometer la arquitectura.

Fotos Evidencias

CREAR BASE DE DATOS

```
1 -- CREAR BASE DE DATOS
2 CREATE DATABASE pawbond;
3 USE pawbond;
```

CREAR TABLAS

```
1 -- CREAR TABLA USUARIOS
2 CREATE TABLE `usuarios` (
3   `idUsuario` INT AUTO_INCREMENT PRIMARY KEY,
4   `nombre` VARCHAR(100) NOT NULL,
5   `correo` VARCHAR(120) UNIQUE,
6   `contraseña` VARCHAR(255) NOT NULL,
7   `telefono` VARCHAR(20) DEFAULT NULL,
8   `numeroDocumento` VARCHAR(20) UNIQUE,
9   `fechaRegistro` DATETIME);|
10
```

```
1 -- ADICIONAR CAMPO DIRECCION A LA TABLA USUARIO
2 ALTER TABLE usuario add COLUMN direccion VARCHAR(50) NOT NULL;
```

```
1 ALTER TABLE usuario
2 MODIFY COLUMN fechaRegistro DATETIME DEFAULT CURRENT_TIMESTAMP;
```

```
1 -- CREAR TABLA MASCOTA
2 CREATE TABLE `mascota` (
3   `idMascota` INT AUTO_INCREMENT PRIMARY KEY,
4   `nombreMascota` VARCHAR(100) NOT NULL,
5   `raza` VARCHAR(50) NOT NULL,
6   `edad` INT NOT NULL,
7   `genero` ENUM('Macho','Hembra'),
8   `especie` VARCHAR(50) NOT NULL,
9   `color` VARCHAR(50) NOT NULL,
10  `fechaRegistroMascota` DATETIME DEFAULT NULL,
11  `idUsuario` INT);
12
```

```
1 ALTER TABLE mascota
2 MODIFY COLUMN fechaRegistroMascota DATETIME DEFAULT CURRENT_TIMESTAMP;
```

```
1 -- CREAR TABLA COLLAR
2 CREATE TABLE `collar` (
3   `idCollar` INT AUTO_INCREMENT PRIMARY KEY,
4   `idMascota` INT NOT NULL,
5   `codigoNFC` INT NOT NULL,
6   `estadoCollar` ENUM('activo','inactivo'));
7
```

```
1 -- CREAR TABLA INFOMEDICA
2 CREATE TABLE `InfoMedica` (
3   `idInfo` INT AUTO_INCREMENT PRIMARY KEY,
4   `idMascota` INT,
5   `peso` TEXT(100) NULL,
6   `medicamentosActuales` TEXT(100) NULL,
7   `alergias` TEXT(100) NULL,
8   `enfermedades` TEXT(100) NULL,
9   `vacunas` TEXT(100) NULL,
10  `fechaUltimaVacuna` DATE DEFAULT NULL,
11  `esterilizado` ENUM ('Si', 'No'),
12  `infoAdicional` TEXT(100) NULL );
```

```
1 CREATE TABLE administrador (  
2     id_administrador INT NOT NULL AUTO_INCREMENT,  
3     nombre VARCHAR(100) NOT NULL,  
4     contrasena VARCHAR(255) NOT NULL,  
5     correo VARCHAR(100) UNIQUE NOT NULL,  
6     documento VARCHAR(20) UNIQUE NOT NULL,  
7     telefono VARCHAR(15),  
8     PRIMARY KEY (id_administrador)  
9 );|
```

```
1 -- CREAR TABLA PUBLICACION|  
2 CREATE TABLE `publicacion` (  
3     `idPublicacion` INT AUTO_INCREMENT PRIMARY KEY,  
4     `idUsuario` INT,  
5     `Comentario` TEXT(120) NOT NULL,  
6     `FechaPublicacion` DATETIME DEFAULT NULL);
```

```
1 ALTER TABLE publicacion  
2 MODIFY COLUMN FechaPublicacion DATETIME DEFAULT CURRENT_TIMESTAMP;
```

```
1 -- CREAR TABLA REPORTE  
2 CREATE TABLE `reporte` (  
3     `idReporte` INT AUTO_INCREMENT PRIMARY KEY,  
4     `idCollar` INT,  
5     `ubicacion` VARCHAR(50) NOT NULL,  
6     `descripcionReporte` TEXT(120) NOT NULL,  
7     `infoContactoReporte` VARCHAR(50) NOT NULL,  
8     `fechaReporte` DATETIME DEFAULT NULL);
```

```
1 ALTER TABLE reporte  
2 MODIFY COLUMN fechaReporte DATETIME DEFAULT CURRENT_TIMESTAMP;
```

LLAVES FORANEAS

```
1 -- LLAVE FORANEA DE USUARIO Y MASCOTA
2 ALTER TABLE mascota ADD
3 CONSTRAINT tiene
4 FOREIGN KEY (idUserario)
5 REFERENCES usuario (idUserario)
```

```
1 -- LLAVE FORANEA DE MASCOTA Y COLLAR
2 ALTER TABLE collar ADD
3 CONSTRAINT pertenece
4 FOREIGN KEY (idMascota)
5 REFERENCES mascota (idMascota)|
6
```

INSERTAR DATOS

```
1 INSERT INTO `usuario` (`nombre`, `correo`, `contraseña`, `telefono`, `numeroDocumento`, `fechaRegistro`, `direccion`)
2 VALUES ('Laura Perez', 'laura@gmail.com', '123456', '3001234567', '10255378954', '2025-02-14', 'Calle 10 # 5-51'),
3 ('Carlitos Gutierrez', 'carlos@gmail.com', 'abc123', '3019876543', '1028695874', '2025-11-14', 'Calle 10 # 5-51'),
4 ('Luisito Comunica', 'luisito@gmail.com', 'abc123', '3019876543', '5228656950', '2024-12-10', 'Carrera 21 # 17-63');|
```

```
1 INSERT INTO `mascota` (`nombreMascota`, `raza`, `edad`, `genero`, `especie`, `color`,
2 `fechaRegistroMascota`, `idUserario`)
3 VALUES('Max', 'Golden Retriever', '4', 'Macho', 'Perro', 'Dorado', '2025-01-25', 5),
4 ('Luna', 'Siamés', '2', 'Hembra', 'Gato', 'Crema y Marrón', '2024-11-10', 6),
5 ('Coco', 'Poodle Toy', '6', 'Hembra', 'Perro', 'Blanco', '2025-03-05', 4);|
```

```
1 -- INSERTAR DATOS COLLAR
2 INSERT INTO `collar` (`idMascota`, `estadoCollar`)
3 VALUES(13,'activo'),
4 (15,'activo'),
5 (14,'inactivo');|
```

```
1 -- INSERTAR DATOS INFORMACION MEDICA
2 INSERT INTO `infoMedica` (`idMascota`, `peso`, `medicamentosActuales`, `alergias`, `enfermedades`, `vacunas`,
3 `fechaUltimaVacuna`, `esterilizado`, `infoAdicional`)
4 VALUES
5 (13, 28.5, 'Ninguno', 'Ninguna conocida', 'Displasia de cadera (leve)', 'Si', '2025-09-15', 'si',
6 'Necesita caminatas cortas; reacciona bien a caricias.'),
7 (15, 4.2, 'Metronidazol', 'Alergia al pollo', 'Problemas digestivos crónicos', 'Si', '2025-07-20', 'si',
8 'Dieta estricta sin proteínas de ave. Muy asustadiza.'),
9 (14, 3.5, 'Ninguno', 'Aspirina', 'Ninguna conocida', 'Si', '2025-10-01', 'no', 'Necesita bebedero bajo. Muy
10 social y juguetona.');
```

```

1  -- INSERTAR PUBLICACIONES
2  INSERT INTO `publicacion` (`idUsuario`, `Comentario`, `FechaPublicacion`)
3  VALUES
4  (4, '¡Alerta! Mi perro Max fue visto cerca del parque central. Collar NFC activo.', '2025-11-15 15:30:00'),
5  (5, 'Encontré un gato Siamés con collar NFC en la Calle 5. Contacté al dueño, estoy esperando respuesta.',
6  '2025-11-15 16:10:00'),
7  (6, 'Busco a mi Poodle Coco. Se perdió hace una hora, por favor escanear el chip si la ven.', '2025-11-15
8  17:45:00');

1  -- INSERTAR REPORTE
2  INSERT INTO `reporte` (`idCollar`, `ubicacion`, `descripcionReporte`, `infoContactoReporte`, `fechaReporte`)
3  VALUES
4  (13, 'Parque Nacional', 'Vi al Golden Retriever en un parque, está tranquilo pero solo. Escaneo realizado con
5  éxito.',
6  'Pedro Gómez (3105551234)',
7  '2025-11-15 16:00:00'),
8  (15, 'La Candelaria', 'Gato Siamés encontrado en un callejón. Parece asustado y desorientado. Dejé comida.', 'Maria
9  Rojas (mrojas@ejemplo.com)', '2025-11-15 17:35:00'),
10 (14, 'Barrio Galerías', 'El Poodle Toy fue visto subiendo a un bus cerca de la estación. Contacto al dueño
    automático.', 'Sistema Automático NFC', '2025-11-15 18:05:00');

```

Diccionario de datos

Nombre Base de datos: pawbond

Nombre tabla: usuario

Descripción: Tabla que guarda los registros de los dueños de las mascotas.

Nombre campo: idUsuario

Tipo dato: INT

Atributos: PRIMARY KEY

Descripción: Identificador único de cada usuario. Es auto incrementable.

Nombre campo: nombre

Tipo dato: VARCHAR

Atributos: NOT NULL

Descripción: Campo que guarda el nombre y apellido correspondiente a cada usuario. Es obligatorio de diligenciar.

Nombre campo: correo

Tipo dato: VARCHAR

Atributos: UNIQUE

Descripción: Campo que guarda el correo de cada usuario. Cada correo es único (UNIQUE), lo que garantiza que no existan valores duplicados en esa columna.

Nombre campo: contraseña

Tipo dato: VARCHAR

Atributos: NOT NULL

Descripción: Campo que guarda la contraseña del usuario para poder ingresar a la aplicación. Es obligatorio de diligenciar.

Nombre campo: teléfono

Tipo dato: INT

Atributos: DEFAULT NUL

Descripción: Campo que guarda el número de teléfono de cada usuario. No es un campo obligatorio de llenar.

Nombre campo: numeroDocumento

Tipo dato: INT

Atributos: UNIQUE

Descripción: Campo que guarda el numero de identificación de cada usuario. Al igual que el correo, es un valor que no puede registrarse más de una vez.

Nombre campo: fechaRegistro

Tipo dato: DATETIME

Atributos: DEFAULT CURRENT_TIMESTAMP

Descripción: Campo que guarda automáticamente la fecha en la que se registra el usuario en la plataforma.

Nombre campo: direccion

Tipo dato: VARCHAR

Atributos: NOT NULL

Descripción: Campo que guarda la dirección del usuario. Es obligatorio de diligenciar.

Nombre tabla: mascota

Descripción: Tabla que guarda los registros de las mascotas correspondientes a cada usuario.

Nombre campo: idMascota

Tipo dato: INT

Atributos: PRIMARY KEY

Descripción: Identificador único de cada mascota. Es auto incrementable.

Nombre campo: nombreMascota

Tipo dato: VARCHAR

Atributos: NOT NULL

Descripción: Campo que guarda el nombre de la mascota. Es obligatorio de diligenciar.

Nombre campo: raza

Tipo dato: VARCHAR

Atributos: NOT NULL

Descripción: Campo que guarda la raza correspondiente a cada mascota. Es obligatorio de diligenciar.

Nombre campo: edad

Tipo dato: INT

Atributos: NOT NULL

Descripción: Campo que guarda la edad correspondiente a cada mascota. Es obligatorio de diligenciar.

Nombre campo: genero

Tipo dato: ENUM('Macho','Hembra')

Atributos: NOT NULL

Descripción: Campo que guarda el género de cada mascota. Solo admite dos respuestas 'Macho o 'Hembra'. Es obligatorio de diligenciar.

Nombre campo: especie

Tipo dato: VARCHAR

Atributos: NOT NULL

Descripción: Campo que guarda la especie de cada mascota (Gato, perro, hámster...). Es obligatorio de diligenciar.

Nombre campo: color

Tipo dato: VARCHAR

Atributos: NOT NULL

Descripción: Campo que guarda el color correspondiente de cada mascota. Es obligatorio de diligenciar.

Nombre campo: fechaRegistroMascota

Tipo dato: DATETIME

Atributos: DEFAULT CURRENT_TIMESTAMP

Descripción: Campo que guarda automáticamente la fecha en la que se registra la mascota en la plataforma.

Nombre campo: idUsuario

Tipo dato: INT

Atributos: FOREIGN KEY

Descripción: Campo que guarda la llave foránea idUsuario que corresponde al dueño de cada mascota.

Nombre tabla: collar

Descripción: Tabla que guarda la información de cada collar asociados a una mascota. Un collar pertenece a una sola mascota.

Nombre campo: idCollar

Tipo dato: INT

Atributos: PRIMARY KEY

Descripción: Identificador único de cada collar. Es auto incrementable.

Nombre campo: idMascota

Tipo dato: INT

Atributos: FOREIGN KEY

Descripción: Campo que guarda la llave foránea idMascota que corresponde a la mascota a la que pertenece el collar.

Nombre campo: estadoCollar

Tipo dato: ENUM ('activo','inactivo')

Atributos: Ninguno

Descripción: Campo que guarda el estado del collar, puede ser 'activo' o 'inactivo'.

Nombre tabla: infoMedica

Descripción: Tabla que guarda la información médica de las mascotas que los usuarios actualizan o que el sistema sincroniza..

Nombre campo: idInfo

Tipo dato: INT

Atributos: PRIMARY KEY

Descripción: Identificador único de cada historial médico de la mascota. Es auto incrementable.

Nombre campo: idMascota

Tipo dato: INT

Atributos: FOREIGN KEY

Descripción: Campo que guarda la llave foránea idMascota que corresponde a la mascota a la que pertenece el historial médico.

Nombre campo: peso

Tipo dato: TEXT

Atributos: NULL

Descripción: Campo que guarda el peso de la mascota. No es obligatorio diligenciar este campo.

Nombre campo: medicamentosActuales

Tipo dato: TEXT

Atributos: NULL

Descripción: Campo que guarda los medicamentos que tome la mascota actualmente. No es obligatorio diligenciar este campo.

Nombre campo: alergias

Tipo dato: TEXT

Atributos: NULL

Descripción: Campo que guarda las alergias de la mascota. No es obligatorio diligenciar este campo.

Nombre campo: enfermedades

Tipo dato: TEXT

Atributos: NULL

Descripción: Campo que guarda las enfermedades que tenga la mascota. No es obligatorio diligenciar este campo.

Nombre campo: vacunas

Tipo dato: TEXT

Atributos: NULL

Descripción: Campo que guarda las vacunas que tenga la mascota. No es obligatorio diligenciar este campo.

Nombre campo: fechaUltimaVacuna

Tipo dato: DATE

Atributos: DEFAULT NULL

Descripción: Campo que guarda la fecha en la que se haya vacunado la mascota por última vez. No es obligatorio diligenciar este campo.

Nombre campo: esterilizado

Tipo dato: ENUM ('si','no')

Atributos: NOT NULL

Descripción: Campo que guarda si la mascota ha sido esterilizada o no. Es obligatorio diligenciar este campo.

Nombre campo: infoAdicional

Tipo dato: TEXT

Atributos: NULL

Descripción: Campo que guarda información adicional de la mascota que el usuario quiera proporcionar. No es obligatorio diligenciar este campo.

Nombre tabla: reporte

Descripción: Tabla que guarda la información médica de las mascotas que los usuarios actualizan o que el sistema sincroniza.

Nombre campo: idReporte

Tipo dato: INT

Atributos: PRIMARY KEY

Descripción: Identificador único de cada reporte de mascotas realizado por los usuarios. Es auto incrementable.

Nombre campo: idCollar

Tipo dato: INT

Atributos: FOREIGN KEY

Descripción: Campo que guarda la llave foránea idCollar para identificar desde cual collar se está haciendo el reporte de perdida.

Nombre campo: ubicación

Tipo dato: VARCHAR

Atributos: NOT NULL

Descripción: Campo que guarda la ubicación donde se encontró a la mascota. Es obligatorio diligenciar este campo.

Nombre campo: descripciónReporte

Tipo dato: TEXT

Atributos: NOT NULL

Descripción: Campo que guarda detalles o la descripción del reporte de la mascota. Es obligatorio diligenciar este campo.

Nombre campo: infoContactoReporte

Tipo dato: VARCHAR

Atributos: NOT NULL

Descripción: Campo que guarda información de contacto de la persona que encontró a la mascota. Es obligatorio diligenciar este campo.

Nombre campo: fechaReporte

Tipo dato: DATETIME

Atributos: DEFAULT CURRENT_TIMESTAMP

Descripción: Campo que guarda automáticamente la fecha en la que se hizo el reporte de la mascota.

Nombre tabla: publicación

Descripción: Tabla que guarda las publicaciones de los usuarios. Permite interacción entre usuarios dentro de la app.

Nombre campo: idPublicacion

Tipo dato: INT

Atributos: PRIMARY KEY

Descripción: Identificador único de cada publicación hecha por los usuarios. Es auto incrementable.

Nombre campo: idUsuario

Tipo dato: INT

Atributos: FOREIGN KEY

Descripción: Campo que guarda la llave foránea idUsuario que corresponde a cada usuario registrado que haga una publicación en el foro.

Nombre campo: comentario

Tipo dato: TEXT

Atributos: NOT NULL

Descripción: Campo que guarda el mensaje o comentario que el usuario desee publicar. Es obligatorio diligenciar este campo.

Nombre campo: fechaPublicacion

Tipo dato: DATETIME

Atributos: DEFAULT CURRENT_TIMESTAMP

Descripción: Campo que guarda automáticamente la fecha en la que el usuario realizó la publicación.