

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

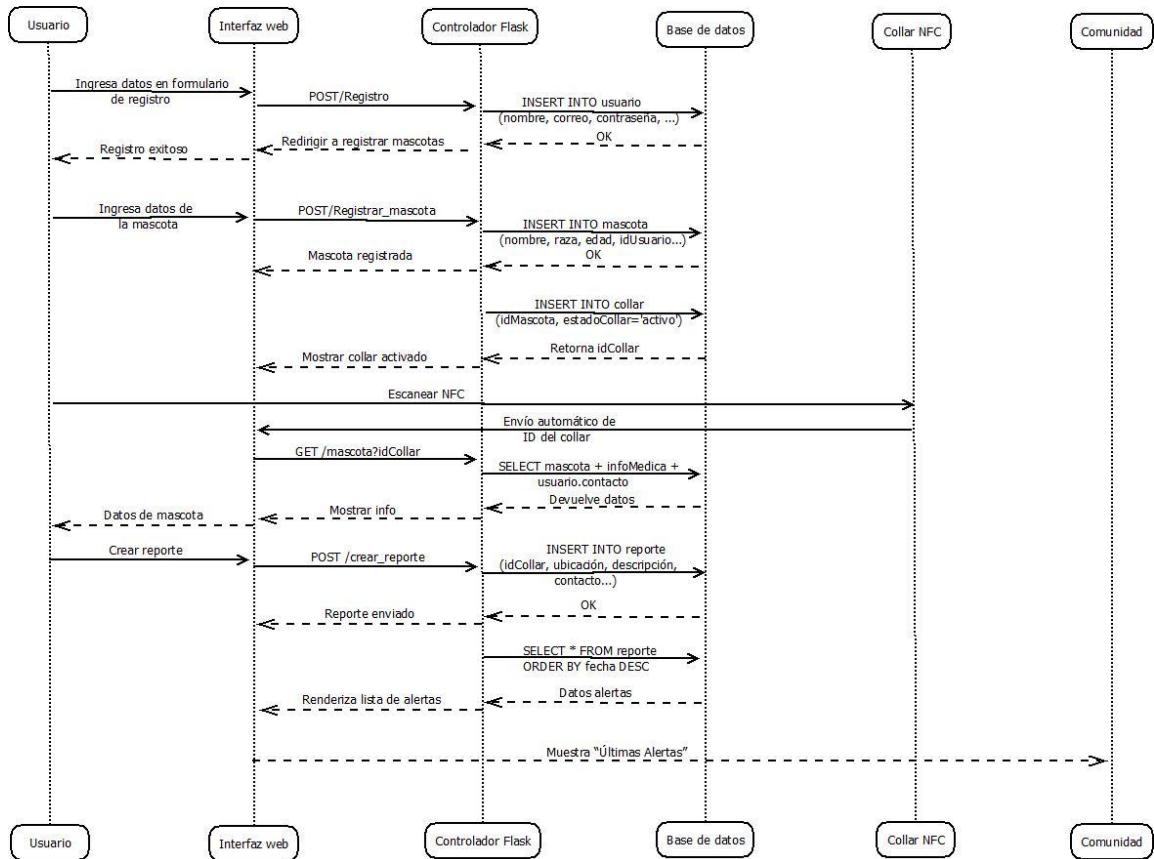
1. Diseñar una plataforma digital en la nube para la gestión de información médica de la mascota y los datos de contacto del dueño.
2. Incorporar un sistema de alertas y notificaciones personalizadas para mejorar la prevención, el cuidado y la localización de mascotas en caso de pérdida.
3. Garantizar 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

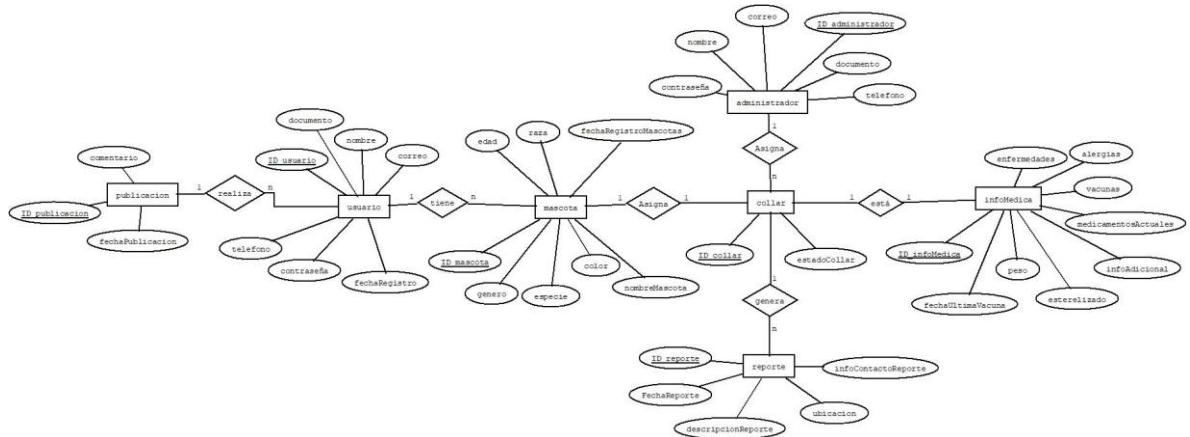
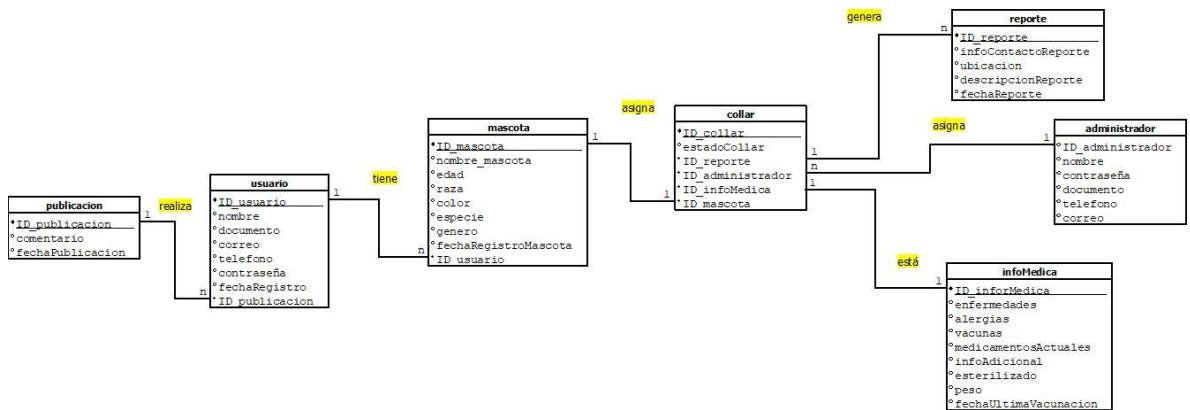
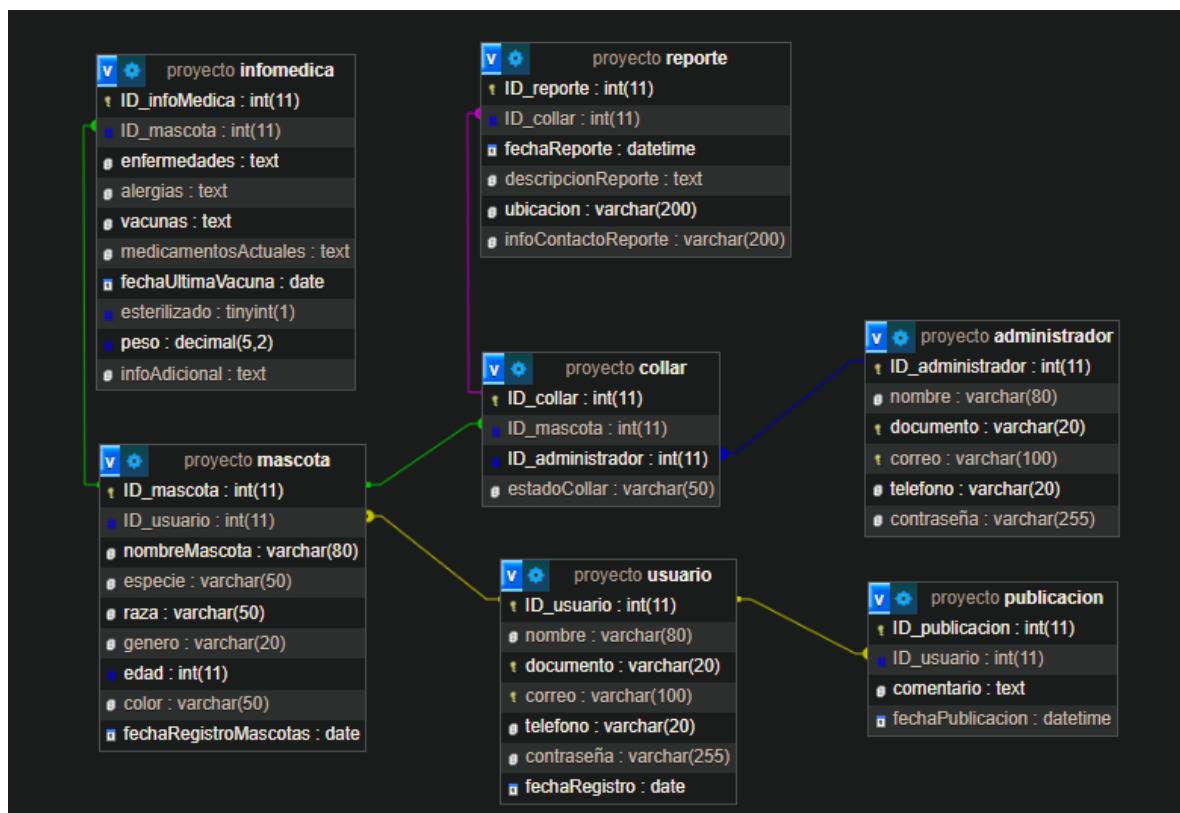


Diagrama Relacional



Modelo Físico



Prototipo Técnico (CRUD Flask + MySQL)

Estructura del proyecto Flask

1. Crear la estructura de carpetas

The screenshot shows a file explorer window with the following structure:

Path	File/Directory	Number
PAWBONDAPP		1
PawBondApp		2
db		3
pawbond.sql		4
static		5
imagenes		6
collar.jpg		7
collar2.png		8
dueño.jpg		9
fundacion.jpg		10
gato.jpg		11
insta.png		12
yutu.png		13
# style.css		14
templates		15
index.html		16
app.py		17
		18
		19

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
11
12 -- ADICIONAR CAMPO DIRECCION A LA TABLA USUARIO
13 ALTER TABLE usuario add COLUMN direccion VARCHAR(50) NOT NULL;
14
15
16 ALTER TABLE usuario
17 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 (idUsuario)
5 REFERENCES usuario (idUsuario)

```

```

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

```

PawBondApp
  db
    pawbond.sql
  static
    imagenes
      collar.jpg
      collar2.png
      dueño.jpg
      fundacion.jpg
      gato.jpg
      insta.png
      yutu.png
    # style.css
  templates
    index.html
  app.py
  BASE DE DATOS PROYECTO.docx
  BASE DE DATOS PROYECTO.pdf
  DIAGRAMA ENTIDAD-RELACION.jpg
  DIAGRAMA RELACIONAL.jpg

```

```

from flask import Flask, render_template, request, redirect, url_for, flash
from flask_mysqldb import MySQL
import datetime

app = Flask(__name__)
app.secret_key = 'mi_clave_secreta_super_segura'

# CONFIGURACIÓN DE LA BASE DE DATOS
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'pawbond'
app.config['MYSQL_CURSORCLASS'] = 'DictCursor'

mysql = MySQL(app)

@app.route('/')
def listar_mascotas():
    mascotas = []
    publicaciones = []

    try:
        cur = mysql.connection.cursor()

        query_mascotas = """
        SELECT
            m.nombreMascota,
            m.especie,
            m.raza,
            m.edad,
            m.genero,
            m.color,
            u.nombre AS nombreDueño,
            u.telefono AS telefonoDueño,
            c.estadoCollar
        FROM
            mascota m
        JOIN
            usuario u ON m.idUsuario = u.idUsuario
        LEFT JOIN
            collar c ON m.idMascota = c.idMascota;
        """

        cur.execute(query_mascotas)
        mascotas = cur.fetchall()

        query_publicaciones = """
        """
        cur.execute(query_publicaciones)
        publicaciones = cur.fetchall()
    except Exception as e:
        flash('Error: ' + str(e))

    return render_template('index.html', mascotas=mascotas, publicaciones=publicaciones)

```

OUTLINE

TIMELINE

4. Desarrollo y funcionamiento del CRUD

- **Crear (Create)**

```
index.html app.py
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
127                 fecha_registro = datetime.datetime.now()
128                 cur.execute(query_usuario, (nombre_u, correo_u, clave_u, doc_u, tel_u, dir_u, fecha_registro))
129
130                 mysql.connection.commit()
131                 cur.close()
132
133                 flash(f'Registro de usuario exitoso! Bienvenido/a {nombre_u}.', 'success')
134
135             except Exception as e:
136                 mysql.connection.rollback()
137                 print(f'Error durante el registro de usuario: {e}')
138                 flash(f'Error al registrar el usuario: {e}', 'error')
139
140             return redirect(url_for('listar_mascotas'))
```

```

142 @app.route('/registrar_mascota', methods=['POST'])
143 def registrar_mascota():
144     if request.method == 'POST':
145         nombre_u = request.form['nombreDueño']
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("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)
171             """
172
173         cur.execute(query_mascota, (id_usuario, nombre_m, raza_m, edad_m, genero_m, especie_m, color_m))
174
175         mysql.connection.commit()
176         cur.close()
177
178         flash(f'{nombre_m} registrada exitosamente!', 'success')
179
180     except Exception as e:
181         mysql.connection.rollback()
182         print(f'Error al registrar mascota: {e}')
183         flash('Error interno al registrar la mascota.', 'error')
184
185     return redirect(url_for('listar_mascotas'))
185

```

```

index.html
1 index.html X app.py
2 PawBonsApp > templates > index.html > index.html
3
4 <html lang="es" style="width: 1000px; margin: 0 auto">
5
6     <!-- registro de mascota -->
7
8     <section id="registro_mascota" style="font-family:Maistrala GD; background-color: #d49b75fc; padding: 30px 0;"><br>
9         <div class="titulos" style="text-align:center;"> Registra tu Mascota:<h2>
10
11         <div class="form_registro_mascota" style="width: 500px; margin: 20px auto; padding: 20px; border: 1px solid #ffbb78c7; border-radius: 8px; background-color: #fffffe;">
12             <form method="POST" action="{{ url_for('registrar_mascota') }}">
13
14                 <h3 style="color: #360181;">Datos del Dueño y Mascota:</h3>
15
16                 <label for="nombreDueño">Tu Nombre (Debe estar registrado):</label>
17                 <input type="text" id="nombreDueño" name="nombreDueño" required style="width: 100%; padding: 8px; margin-bottom: 20px; box-sizing: border-box;">
18
19                 <label for="nombreMascota">Nombre de la Mascota:</label>
20                 <input type="text" id="nombreMascota" name="nombreMascota" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
21
22                 <label for="especie">Especie:</label>
23                 <select id="especie" name="especie" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
24                     <option value="Perro">Perro</option>
25                     <option value="Gato">Gato</option>
26                     <option value="Otro">Otro</option>
27                 </select>
28
29                 <label for="raza">Raza:</label>
30                 <input type="text" id="raza" name="raza" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
31
32                 <label for="edad">Edad (Años):</label>
33                 <input type="number" id="edad" name="edad" min="0" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
34
35                 <label for="genero">Genero:</label>
36                 <select id="genero" name="genero" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
37                     <option value="Hacho">Hacho</option>
38                     <option value="Hembra">Hembra</option>
39                 </select>
40
41                 <label for="color">Color:</label>
42                 <input type="text" id="color" name="color" required style="width: 100%; padding: 8px; margin-bottom: 10px; box-sizing: border-box;">
43
44                 <label for="esterilizado">¿Está esterilizado? (Validación simple):</label>
45                 <select id="esterilizado" name="esterilizado" style="width: 100%; padding: 8px; margin-bottom: 20px; box-sizing: border-box;">
46                     <option value="No">No</option>
47                     <option value="Si">Sí</option>
48                 </select>
49
50                 <button type="submit" style="background-color: #360181; color: white; padding: 10px 15px; border: none; border-radius: 5px; cursor: pointer; width: 100%;">Registrar Mascota</button>
51             </form>
52         </div>
53         <br>
54         <br>
55     </section>
56

```

- (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
93
94         cur.execute(query_publicacion, (id_usuario, comentario))
95
96         mysql.connection.commit()
97         cur.close()
98
99         flash('¡Alerta publicada exitosamente!', 'success')
100
101     except Exception as e:
102         mysql.connection.rollback()
103         print(f"Error al publicar alerta: {e}")
104         flash('Error interno al publicar la alerta.', 'error')
105
106
107     return redirect(url_for('listar_mascotas'))
108
109
110 @app.route('/')
111 def listar_mascotas():
112     mascotas = []
113     publicaciones = []
114
115     try:
116         cur = mysql.connection.cursor()
117
118         query_mascotas = """
119             SELECT
120                 m.nombreMascota,
121                 m.especie,
122                 m.raza,
123                 m.edad,
124                 m.genero,
125                 m.color,
126                 u.nombre AS nombreDueño,
127                 u.telefono AS telefonoDueño,
128                 c.estadoCollar
129             FROM
130                 mascota m
131             JOIN
132                 usuario u ON m.idUsuario = u.idUsuario
133             LEFT JOIN
134                 collar c ON m.idMascota = c.idMascota;
135             """
136
137         cur.execute(query_mascotas)
138         mascotas = cur.fetchall()
139
140         query_publicaciones = """
141             SELECT
142                 p.Comentario,
143                 p.FechaPublicacion,
144                 u.nombre AS nombreUsuario
145             FROM
146                 publicacion p
147             JOIN
148                 usuario u ON p.idUsuario = u.idUsuario
149             ORDER BY
150                 p.FechaPublicacion DESC;
151             """
152
153         cur.execute(query_publicaciones)
154         publicaciones = cur.fetchall()
155
156         cur.close()
157
158     except Exception as e:
159         print(f"Error al listar datos: {e}")
160         flash('Error al cargar datos de la base de datos.', 'error')
161
162
163     return render_template('index.html', mascotas=mascotas, publicaciones=publicaciones)
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
627
628
629
629
630
631
632
633
634
635
635
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1
```

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

The screenshot shows a web browser window with multiple tabs open. The active tab displays a registration form titled "Prueba inicial del registro de alertas". The form is for a pet owner (Dueño) and includes fields for Name, Email, Password, Document Number, Phone, and Address. A success message at the top of the form states: "(Así se haría un registro de usuario/mascota, ya estando registro podemos hacer alertas)".

Nombre Completo:	Jose Maria Cordoba
Correo Electrónico:	joseito@gmail.com
Contraseña:	*****
Número de Documento:	1234567890
Teléfono:	3002345794
Dirección:	calle 14 sur
<input type="button" value="Registrar Usuario"/>	

The screenshot shows a MySQL database interface with a sidebar containing schema names like 'pawbond', 'performance_schema', 'phpmyadmin', and 'test'. The main area displays a table named 'usuario' with columns: idUsuario, nombre, correo, contraseña, telefono, numeroDocumento, fechaRegistro, and direccion. The table contains six rows of data. The last row, which corresponds to the user registered in the previous screenshot, is highlighted with a yellow background.

	Editar	Copiar	Borrar	4	Laura Perez	laura@gmail.com	123456	2147483647	10255378954	2025-02-14 00:00:00	Calle 10 # 5-51
	Editar	Copiar	Borrar	5	Carlitos Gutierrez	carlos@gmail.com	abc123	2147483647	1028695674	2025-11-14 00:00:00	Calle 10 # 5-51
	Editar	Copiar	Borrar	6	Luisito Comunica	luisito@gmail.com	abc123	2147483647	5228656950	2024-12-10 00:00:00	Carrera 21 # 17-63
	Editar	Copiar	Borrar	7	José María Córdoba	joseito@gmail.com	123456789	2147483647	12548655789	2025-11-25 21:06:04	calle 14 sur

iCrea 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 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

 | 
Click siquieres 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 Maria 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."

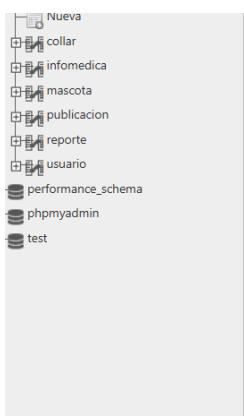
Publicado por Luisito Comunica el 2025-11-15 17:45:00

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

Publicado por Carlos 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



MySQL Database Structure:

- Nueva
- collar
- infomedica
- mascota
- publicacion
- reporte
- usuario
- performance_schema
- phpmyadmin
- test

Alerts Table Data:

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

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 (idUsuario)
5 REFERENCES usuario (idUsuario)
```

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

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`, `idUsuario`)
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
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.',
'2025-11-15 16:10:00'),
6 (6, 'Busco a mi Poodle Coco. Se perdió hace una hora, por favor escanear el chip si la ven.', '2025-11-15
17:45:00');

1 -- InSERTAR REPORTES
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
éxito.',
5     'Pedro Gómez (3105551234)',
6     '2025-11-15 16:00:00'),
7
8 (15, 'La Candelaria', 'Gato Siamés encontrado en un callejón. Parece asustado y desorientado. Dejé comida.', 'Maria
Rojas (mrojas@ejemplo.com)', '2025-11-15 17:35:00'),
9
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: FOREING KEY

Descripción: Campo que guarda la llave foranea 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: FOREING 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: FOREING 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: FOREING 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.