Universidad Autónoma Tomás Frías
Facultad de Ciencias Puras
Carrera de Ingeniería Informática
Desarrollo de Aplicaciones Móviles (INF 632)

Proyecto No 3 (Trabajo Final)

Fecha de publicación: 13/06/2025

Fecha de presentación: 04/07/2025

Formato de presentación: Presentarlo en su dispositivo móvil para revisión en horario de clases. Posterior a recibir las recomendaciones subir el proyecto a GitHub y entregarlo mediante la plataforma virtual de la UATF.

Proyecto Avanzado Flutter: "VetCare Pro"

Registro de Pacientes y Gestión de Historias Clínicas con Persistencia de Datos

Objetivo

Desarrollar una aplicación Flutter funcional para una veterinaria, que permita **registrar pacientes (mascotas)**, gestionar **historias clínicas**, y **guardar los datos de forma persistente** usando alguna de las siguientes opciones:

- sqflite (base de datos local SQLite)
- API REST propia o externa
- Firebase (Firestore y/o Realtime Database)

Modalidades del Proyecto (elige 1 para implementar)

Modalidad	Descripción breve
Local (sqflite)	Todos los datos se guardan y consultan desde SQLite en el dispositivo. Ideal para apps offline.
Remota (API REST)	Los datos se consumen desde un servidor vía HTTP (http, dio, etc.). Ideal para trabajar con backend.
Firebase	Los datos se guardan en Firestore (o Realtime DB) con autenticación opcional. Ideal para apps conectadas a la nube.

Funcionalidades Comunes

1. Pantalla Principal (HomePage)

- Muestra una lista de pacientes.
- Cada paciente:
    - Nombre, especie y fecha de nacimiento.
    - Botón para ver detalles clínicos.
- Botón flotante para registrar nueva mascota.

2. Registro de Pacientes (RegisterPatientPage)

- Formulario validado con campos:
    - Nombre, especie, raza, fecha de nacimiento, propietario.

3. Historia Clínica (MedicalRecordPage)

- Lista de entradas clínicas del paciente (fecha + descripción).
- Botón para **añadir entrada**.
- Botón para **eliminar** entradas.
- Las entradas están asociadas a un paciente.

Gestión de Estado

- Usa Provider para gestionar:
    - Lista de pacientes.
    - Historia clínica por paciente.
    - Eventos como registrar, eliminar, actualizar.

Persistencia de Datos

Con sqflite

- Modelo Patient y MedicalRecord se almacenan en tablas separadas.
- Uso de claves foráneas.
- Consultas con JOIN o por patientId.

Con API REST

- CRUD a través de endpoints:
    - /patients, /patients/:id/records
- Requiere http o dio.
- Simulación: usar json-server o Node.js + Express.

Con Firebase

- Datos en Firestore:

    - Colección patients
        - Subcolección medicalRecords
- Autenticación con email (opcional).
- Uso de firebase_core, cloud_firestore.

Lista de Cotejo - Proyecto "VetCare Pro"

Nombre del Alumno:
Modalidad elegida (marca una):
☐ sqflite ☐ API REST ☐ Firebase

1. Funcionalidad General

No.	Criterio	Cumple (✓)	Observaciones
1	Se pueden registrar, listar y eliminar pacientes		
2	Se pueden registrar, listar y eliminar entradas clínicas		
3	La información se muestra correctamente entre pantallas		
4	La app mantiene los datos tras cerrar sesión o reiniciar (persistencia)		

2. Persistencia de Datos

No.	Criterio	Cumple (✓)	Observaciones
5	Se guarda y recupera la lista de pacientes correctamente		
6	Se guarda y recupera la historia clínica asociada al paciente		
7	Operaciones CRUD funcionan correctamente (guardar, editar, eliminar)		
8	Manejo de errores en carga/sincronización (snackbar, alerts, etc.)		

3. Código y Buenas Prácticas

No.	Criterio	Cumple (✓)	Observaciones
9	Uso correcto de ChangeNotifier, Consumer, FutureBuilder/StreamBuilder		
10	Código estructurado por modelos, servicios, pantallas y widgets		
11	Validación de formularios y datos obligatorios		
12	Uso de UUID, fechas y relaciones correctas entre entidades		

4. Extras (Opcionales)

No.	Criterio	Cumple (✓)	Observaciones
13	Pantalla de búsqueda o filtrado de pacientes		
14	Edición de datos del paciente		
15	Autenticación de usuarios (Firebase Auth o JWT)		

Evaluación Final

- **Mínimo aprobatorio:** 10 ✓ en criterios 1–12.
- **Cada ✓ en extras (13–15)** suma puntos adicionales.

Total de ✓ (Obligatorios): ____ / 12
Total de ✓ (Extras): ____ / 3
NOTA FINAL: _____