

Universidad Autónoma Tomás Frías
Facultad de Ciencias Puras
Carrera de Ingeniería Informática

Asignatura: Desarrollo de Aplicaciones Móviles (INF 662)

Docente: M. Sc. Huáscar Fedor Gonzales Guzmán

TRABAJO DE SEGUNDO PARCIAL

Fecha de publicación: 21/11/2025

Fecha de presentación: 24/11/2025

Formato de presentación:

- Mostrar el proyecto en su **dispositivo móvil** en horario de clases.
- Presentar **impresa la lista de cotejo** para la evaluación.

“ContactApp: Agenda telefónica con Flutter, Provider y SQLite (sqflite)”

Desarrollar una aplicación móvil que permita gestionar una **agenda de contactos telefónicos**, donde el usuario pueda **registrar, listar, buscar, editar y eliminar contactos**, almacenando los datos localmente en una base de datos **SQLite** utilizando el paquete **sqflite**, y gestionando el estado de la aplicación con **Provider**.

1. Objetivos de aprendizaje

Al finalizar el proyecto, el estudiante deberá ser capaz de:

1. Crear una aplicación Flutter estructurada en capas (modelos, servicios, proveedores, UI).
2. Utilizar **Provider** para:
 - Manejar el estado de la lista de contactos.
 - Notificar cambios a la interfaz de usuario.
3. Implementar **persistencia local** con **sqflite**:
 - Crear y actualizar tablas.
 - Ejecutar operaciones CRUD (Create, Read, Update, Delete).
4. Diseñar interfaces simples y usables para formularios y listados.
5. Aplicar buenas prácticas básicas:
 - Separación de responsabilidades.
 - Uso de ChangeNotifier y ChangeNotifierProvider.

2. Requisitos técnicos

- **Gestor de estado:** Provider.
- **Base de datos local:** sqflite + path_provider.
- **Mínimo de pantallas:**
 1. Pantalla de listado de contactos.
 2. Pantalla de detalles del contacto.
 3. Pantalla de creación/edición de contacto.

3. Funcionalidades mínimas

3.1. Alta de contactos

- Formulario con al menos los siguientes campos:
 - Nombre completo (obligatorio).
 - Teléfono principal (obligatorio).
 - Teléfono secundario (puede recibir vacío).
 - Correo electrónico (puede recibir vacío).
 - Nota / descripción (puede recibir vacío).
- Botón **Guardar:**
 - Si es un contacto nuevo → lo inserta en la base de datos.
 - Si es un contacto existente → lo actualiza.
- Validaciones básicas:
 - Nombre y teléfono principal no vacío.
 - Teléfono con longitud mínima razonable (ej. ≥ 6 caracteres).

3.2. Listado de contactos

- Mostrar la lista de contactos ordenados alfabéticamente por nombre.
- Cada ítem debe mostrar:
 - Nombre del contacto.
 - Teléfono principal.
- Al tocar un contacto, navegar a una pantalla de **detalles**.

3.3. Edición de contactos

- A partir de la pantalla de detalles, permitir **editar** los datos.
- Los cambios se guardan tanto en la interfaz como en SQLite.
- Mostrar feedback visual (ej. Snackbar indicando “Contacto actualizado”).

3.4. Eliminación de contactos

- Desde el listado o detalle:
 - Opción de eliminar contacto (por icono de papelera).
 - Mostrar un diálogo de confirmación: “¿Desea eliminar este contacto?”

- Al confirmar:
 - Eliminar de SQLite.
 - Actualizar la lista (Provider).

3.5. Búsqueda de contactos

- En la pantalla de listado:
 - Un campo de búsqueda (AppBar con Search o TextField).
 - Filtrar contactos por nombre en tiempo real a través de Provider.

4. Extensiones opcionales (puntos extra)

- Marcador de **favoritos** (atributo isFavorite en Contact).
- Soporte para **foto** del contacto (almacenada como ruta de archivo o base64).
- Exportar lista de contactos a un archivo (ej. JSON).
- Implementar **paginación** o lazy loading si la lista crece mucho.
- Ordenar por nombre, o por favoritos primero.
- Tema claro/oscuro.

Lista de cotejo – App de Contactos con SQLite + Provider

Apellidos y nombres:

A. Funcionalidades mínimas (OBLIGATORIAS)

3.1 Alta de contactos

Ítem	Criterio de evaluación	Sí	No	Observaciones
3.1.1	El formulario de alta/edición incluye el campo Nombre completo .			
3.1.2	El campo Nombre completo es obligatorio (validación de no vacío).			
3.1.3	El formulario incluye el campo Teléfono principal .			
3.1.4	El campo Teléfono principal es obligatorio (validación de no vacío).			
3.1.5	El formulario incluye el campo Teléfono secundario .			
3.1.6	El campo Teléfono secundario acepta estar vacío sin errores en la app.			
3.1.7	El formulario incluye el campo Correo electrónico .			
3.1.8	El campo Correo electrónico acepta estar vacío sin errores en la app.			
3.1.9	El formulario incluye el campo Nota / descripción .			
3.1.10	El campo Nota / descripción acepta estar vacío sin errores en la app.			
3.1.11	Existe un botón Guardar claramente visible.			
3.1.12	Al guardar un contacto nuevo , éste se inserta en la base de datos SQLite correctamente.			
3.1.13	Al guardar un contacto existente , los datos se actualizan en la base de datos SQLite correctamente.			
3.1.14	Se valida que Nombre no esté vacío antes de guardar.			
3.1.15	Se valida que Teléfono principal no esté vacío antes de guardar.			
3.1.16	El Teléfono principal exige una longitud mínima razonable (p.ej. ≥ 6 caracteres).			
3.1.17	Ante errores de validación, la app muestra mensajes claros (ej. texto de error debajo del campo o Snackbar).			

3.2 Listado de contactos

Ítem	Criterio de evaluación	Sí	No	Observaciones
3.2.1	Existe una pantalla principal que muestra un listado de contactos .			
3.2.2	Los contactos del listado se muestran ordenados alfabéticamente por nombre .			
3.2.3	Cada ítem de la lista muestra claramente el nombre del contacto .			
3.2.4	Cada ítem de la lista muestra claramente el teléfono principal del contacto.			
3.2.5	Al tocar un contacto del listado, la app navega a una pantalla de detalles de ese contacto.			

3.3 Edición de contactos

Ítem	Criterio de evaluación	Sí	No	Observaciones
3.3.1	Desde la pantalla de detalles existe una opción clara para editar el contacto (botón, ícono, etc.).			

3.3.2	Al seleccionar editar, se muestra un formulario con los datos actuales del contacto precargados.			
3.3.3	Los cambios realizados en el formulario se guardan correctamente en SQLite al confirmar.			
3.3.4	Los cambios se reflejan inmediatamente en la interfaz (lista y/o detalles) tras guardar.			
3.3.5	La app muestra feedback visual al actualizar (ej. Snackbar “Contacto actualizado” u otra notificación equivalente).			

3.4 Eliminación de contactos

Ítem	Criterio de evaluación	Sí	No	Observaciones
3.4.1	Desde el listado existe una opción clara para eliminar un contacto (por ejemplo, ícono de papelera).			
3.4.2	Desde la pantalla de detalles existe también una opción para eliminar el contacto (si se definió así).			
3.4.3	Antes de eliminar, se muestra un diálogo de confirmación con el mensaje: “¿Desea eliminar este contacto?” (o similar).			
3.4.4	Al confirmar la eliminación, el contacto se elimina de SQLite correctamente.			
3.4.5	Tras eliminar, la lista se actualiza a través de Provider (el contacto eliminado ya no aparece).			

3.5 Búsqueda de contactos

Ítem	Criterio de evaluación	Sí	No	Observaciones
3.5.1	En la pantalla de listado existe un campo de búsqueda (en AppBar o mediante un TextField visible).			
3.5.2	La búsqueda filtra los contactos por nombre de forma correcta.			
3.5.3	El filtrado de contactos sucede en tiempo real mientras el usuario escribe.			
3.5.4	El filtrado se realiza a través de Provider u otro mecanismo de gestión de estado equivalente (no solo filtrado local sin estado).			

B. Extensiones opcionales (PUNTOS EXTRA)

Ítem	Criterio de evaluación	Sí	No	Observaciones
4.1	La entidad Contact incluye un atributo isFavorite y la app permite marcar/desmarcar contactos como favoritos.			
4.2	La interfaz ofrece una forma clara de distinguir contactos favoritos (ícono, sección, color, etc.).			
4.3	La app soporta foto del contacto (almacenada como ruta de archivo o en base64) y la muestra en la interfaz.			
4.4	Existe una opción para exportar la lista de contactos a un archivo (ej. JSON) de forma funcional.			
4.5	Se implementa algún mecanismo de paginación o lazy loading cuando la lista de contactos es extensa.			
4.6	La app permite ordenar la lista de contactos (por nombre, y/o favoritos primero).			
4.7	La app incorpora un tema claro/oscuro , seleccionable o dependiente de la configuración del sistema.			