

Requerimientos Técnicos - Proyectos para Lenguajes de Programación

Cada uno de estos proyectos debe aplicar:

- Manejo de estado con **GetX**
 - Persistencia local con **GetStorage** (inicialmente)
 - Migración opcional a **Firestore** para almacenamiento remoto (Segunda etapa)
 - Navegación con **go_router** o **Navigator 1.0**
 - Consumo de APIs simuladas (se puede usar JSON Server o MockAPI)
 - Validaciones de formularios.
 - Uso adecuado de `TextEditingController`
-

Proyecto 1: App de Gestión de Medicación Diaria

Descripción

Aplicación que permite a los usuarios registrar sus medicamentos, configurar recordatorios diarios y marcar dosis como tomadas.

Funcionalidades requeridas

- Registro e inicio de sesión local (usando GetStorage)
- CRUD de medicamentos (nombre, dosis, frecuencia, horarios)
- Notificaciones locales para recordatorios (opcional, paquete: `flutter_local_notifications`)
- Marcar dosis como tomada
- Listado de historial de toma diaria
- Ver información del medicamento

Componentes técnicos clave

- Listas reactivas con `Obx`
 - Uso de `ListView.builder`
 - Formularios validados
 - Persistencia local con GetStorage
-

Proyecto 2: App de Reservación de Canchas Deportivas

Descripción

Aplicación que permite a usuarios ver canchas disponibles, reservar horarios y revisar sus reservaciones.

Funcionalidades requeridas

- Registro e inicio de sesión
- Ver listado de canchas disponibles (nombre, tipo, ubicación)
- Ver horarios disponibles
- Reservar una cancha en un horario específico
- Validación para evitar dobles reservas
- Historial de reservaciones

Componentes técnicos clave

- Manejo de fechas y horas
 - Uso de calendarios o `DatePicker`
 - Manejo de estado para validar disponibilidad
 - Almacenamiento en `GetStorage` inicialmente
-

Proyecto 3: App de Control de Gastos Personales

Descripción

Aplicación que permite a los usuarios registrar sus ingresos y gastos, categorizar movimientos y ver estadísticas simples.

Funcionalidades requeridas

- Registro e inicio de sesión local
- Agregar ingresos y egresos (monto, categoría, fecha)
- Visualización de lista de movimientos
- Filtros por fecha y categoría
- Estadísticas (suma total ingresos/egresos, gráfica de barras o circular)

Componentes técnicos clave

- Uso de paquetes de gráficos (como `fl_chart`)
 - Manejo de filtros con estados `GetX`
 - Control de fechas
 - Guardado local con `GetStorage`
-

Proyecto 4: App de Registro y Seguimiento de Tareas

Universitarias

Descripción

Aplicación enfocada a estudiantes para registrar tareas, asignarlas por asignatura y marcar el avance de cada una.

Funcionalidades requeridas

- Registro e inicio de sesión
- CRUD de asignaturas
- CRUD de tareas por asignatura (título, descripción, fecha de entrega, estado)
- Vista de tareas por asignatura o por semana
- Marcar tareas como completadas
- Notificaciones opcionales de recordatorio (antes de vencimiento)

Componentes técnicos clave

- Manejo de fechas con `DateTime` y `DatePicker`
- Vista de calendario o lista semanal
- Filtros y cambios de estado reactivo con `GetX`
- Persistencia local inicial, migrable a Firebase

Nota: Para la entrega:

- Repositorio del proyecto donde se vean las participaciones de todos los miembros del equipo (**todo deben participar, de lo contrario, no tendrá calificación quien no tenga aportes significativos**)
- README con la descripción del app y los integrantes del equipo