

**UI (Screens/Widgets):** Contiene las pantallas (como el listado y el detalle) y los componentes visuales (tarjetas, botones). Se encarga de mostrar la información al usuario y capturar sus acciones.

**Logic de negocio y estado:** Utilizamos el patrón Provider con ChangeNotifier. Esta clase actúa como el "cerebro" de la app; guarda las listas de categorías y tareas, y contiene las funciones para crear, editar o eliminar datos.

**Flujo:** El usuario toca un botón -> Se llama una función en el Provider -> El Provider modifica la lista y llama a notifyListeners() -> La UI se reconstruye automáticamente.

**Modelo de Datos:** Son clases (TodoCategory y TodoItem) que definen qué información tiene una categoría o una tarea (ID, título, fecha, etc.) sin contener lógica compleja.

### **Flujo General de la Aplicación:**

1. El **Usuario** interactúa con la interfaz (por ejemplo, presiona el botón "Guardar" para una nueva tarea).
2. La **Interfaz (UI)** llama a un método del TodoProvider (ej. addTodo()).
3. El **TodoProvider** actualiza sus listas internas (agrega la tarea a la categoría correspondiente).
4. El Provider ejecuta el método notifyListeners().
5. Flutter detecta el cambio y la UI se redibuja automáticamente para mostrar la lista actualizada.

## Diagrama de Arquitectura

