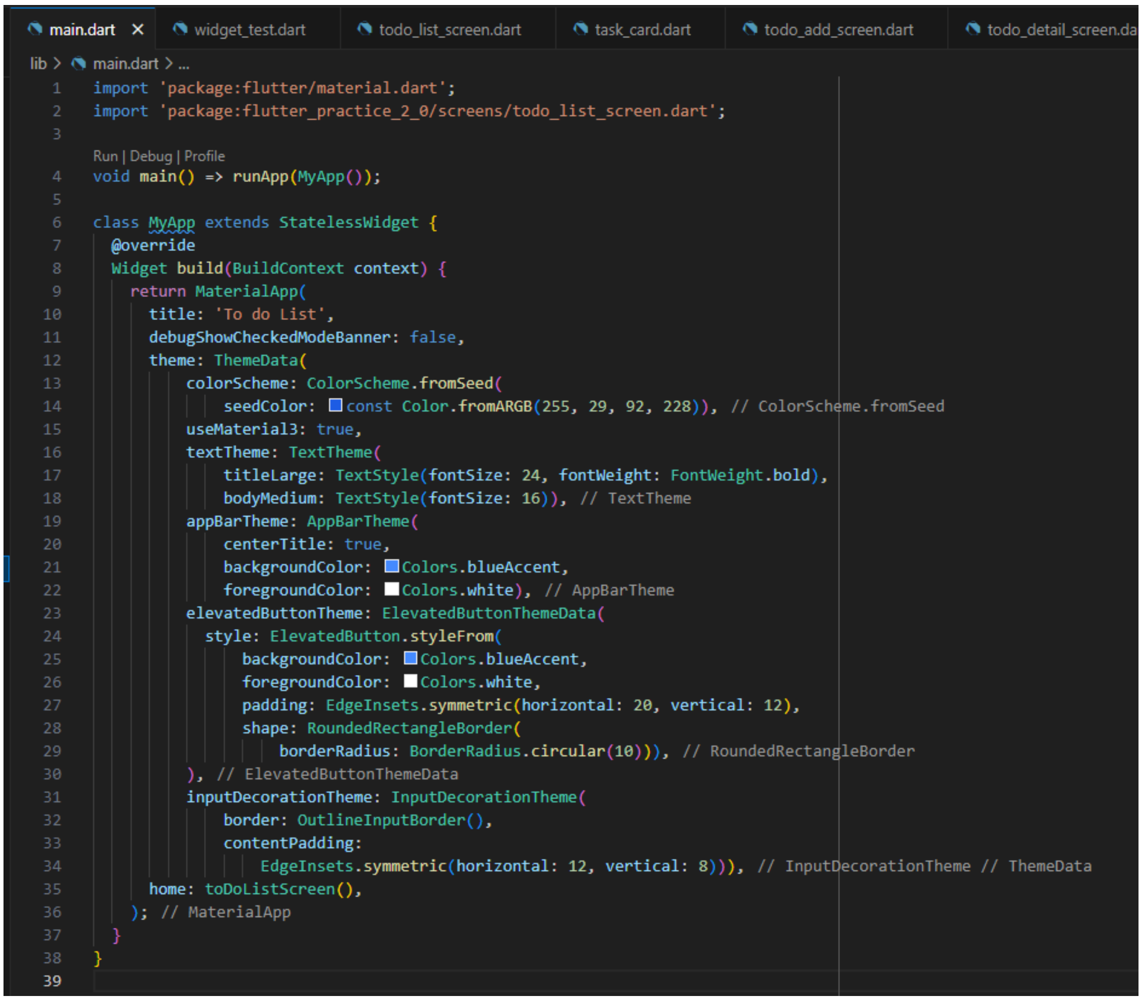
Settings para una app bonita de base



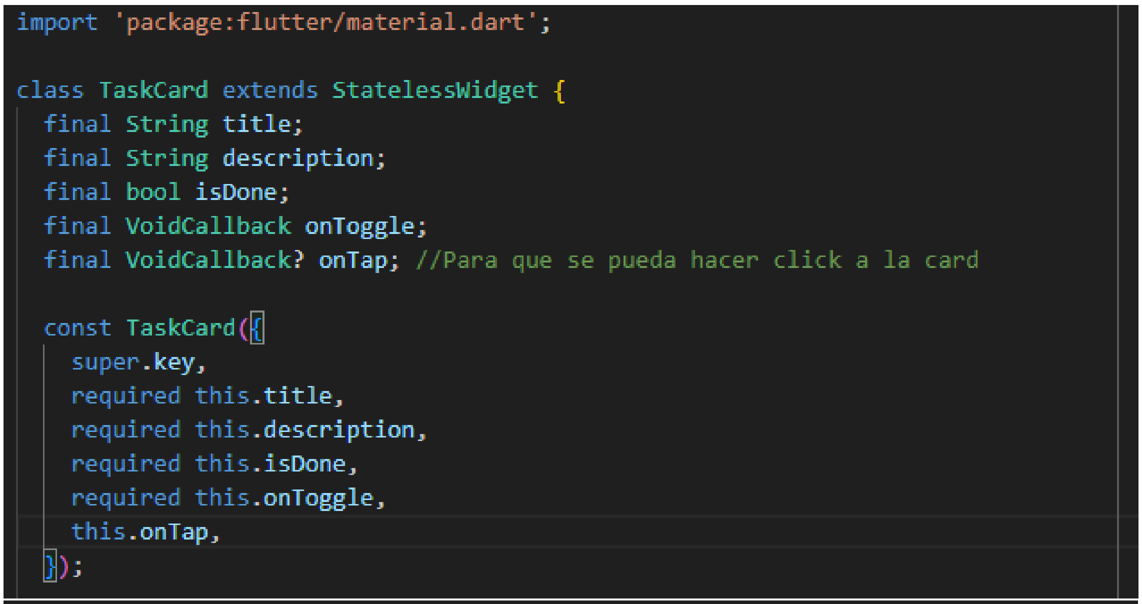
main.dart ✕ | widget_test.dart | todo_list_screen.dart | task_card.dart | todo_add_screen.dart | todo_detail_screen.da

lib > main.dart > ...

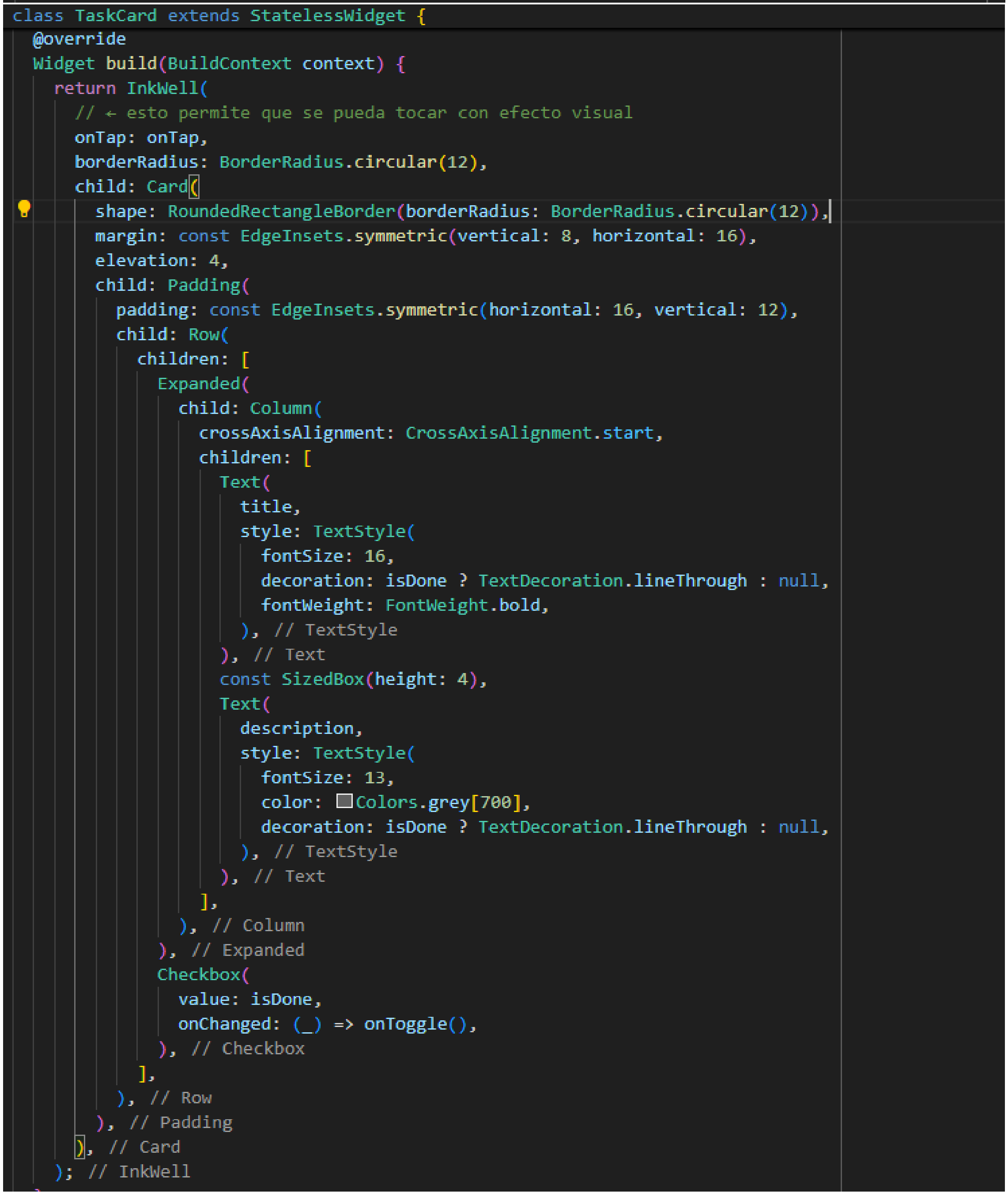
```dart
import 'package:flutter/material.dart';
import 'package:flutter_practice_2_0/screens/todo_list_screen.dart';

Run | Debug | Profile
void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'To do List',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(
          seedColor: const Color.fromARGB(255, 29, 92, 228)), // ColorScheme.fromSeed
        useMaterial3: true,
        textTheme: TextTheme(
          titleLarge: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
          bodyMedium: TextStyle(fontSize: 16)), // TextTheme
        appBarTheme: AppBarTheme(
          centerTitle: true,
          backgroundColor: Colors.blueAccent,
          foregroundColor: Colors.white), // AppBarTheme
        elevatedButtonTheme: ElevatedButtonThemeData(
          style: ElevatedButton.styleFrom(
            backgroundColor: Colors.blueAccent,
            foregroundColor: Colors.white,
            padding: EdgeInsets.symmetric(horizontal: 20, vertical: 12),
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(10))), // RoundedRectangleBorder
        ), // ElevatedButtonThemeData
        inputDecorationTheme: InputDecorationTheme(
          border: OutlineInputBorder(),
          contentPadding:
            EdgeInsets.symmetric(horizontal: 12, vertical: 8))), // InputDecorationTheme // ThemeData
      home: toDoListScreen(),
    ); // MaterialApp
  }
}

```

IMPORTANTE RECORDAR: import ‘package:flutter/material.dart’;

Una card:

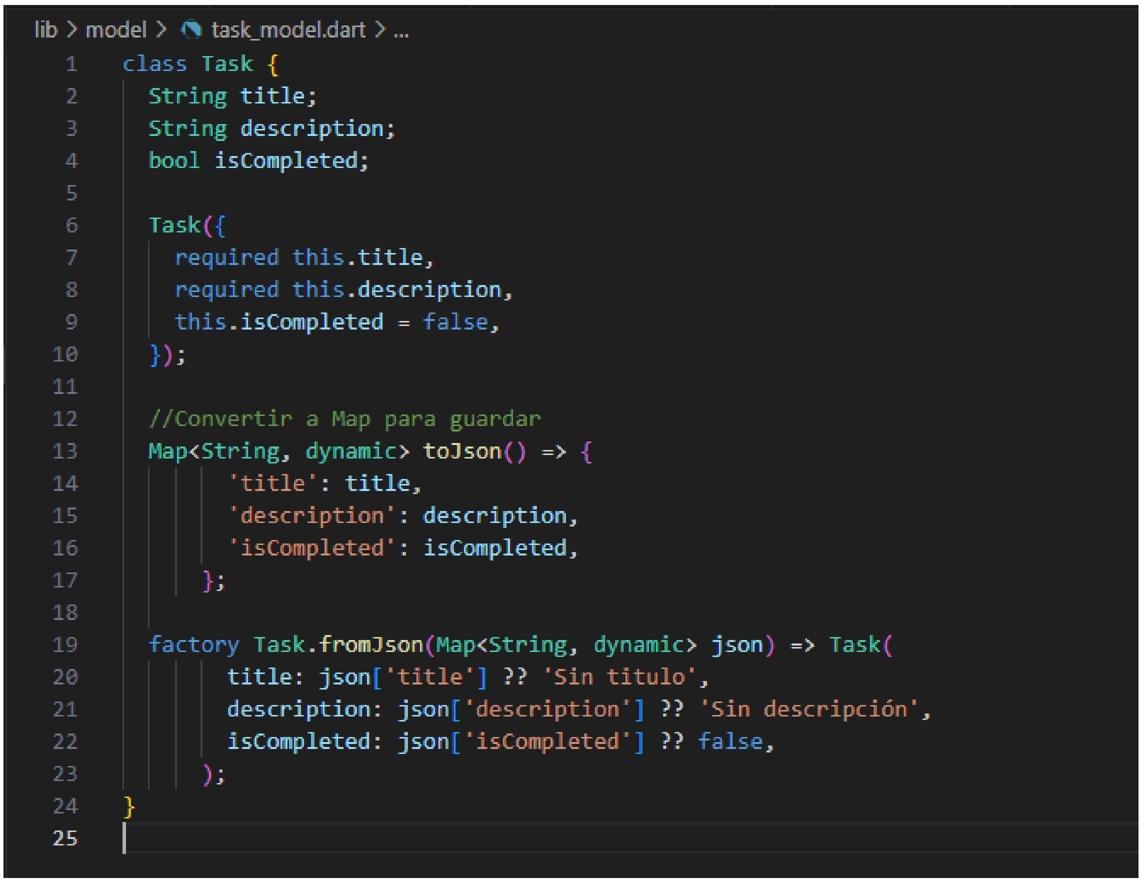


```dart
import 'package:flutter/material.dart';

class TaskCard extends StatelessWidget {
  final String title;
  final String description;
  final bool isDone;
  final VoidCallback onToggle;
  final VoidCallback? onTap; //Para que se pueda hacer click a la card

  const TaskCard({
    super.key,
    required this.title,
    required this.description,
    required this.isDone,
    required this.onToggle,
    this.onTap,
  });
```



```dart
class TaskCard extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return InkWell(
      // ← esto permite que se pueda tocar con efecto visual
      onTap: onTap,
      borderRadius: BorderRadius.circular(12),
      child: Card(
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),
        margin: const EdgeInsets.symmetric(vertical: 8, horizontal: 16),
        elevation: 4,
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 12),
          child: Row(
            children: [
              Expanded(
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      title,
                      style: TextStyle(
                        fontSize: 16,
                        decoration: isDone ? TextDecoration.lineThrough : null,
                        fontWeight: FontWeight.bold,
                      ), // TextStyle
                    ), // Text
                    const SizedBox(height: 4),
                    Text(
                      description,
                      style: TextStyle(
                        fontSize: 13,
                        color: Colors.grey[700],
                        decoration: isDone ? TextDecoration.lineThrough : null,
                      ), // TextStyle
                    ), // Text
                  ],
                ), // Column
              ), // Expanded
              Checkbox(
                value: isDone,
                onChanged: (_) => onToggle(),
              ), // Checkbox
            ],
          ), // Row
        ), // Padding
      ), // Card
    ); // InkWell
```

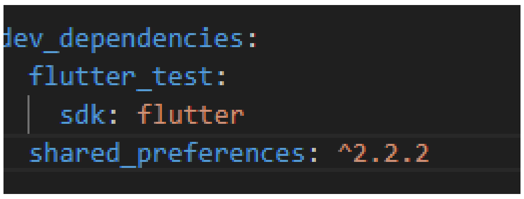
Para la persistencia en model:



lib > model > task_model.dart > ...

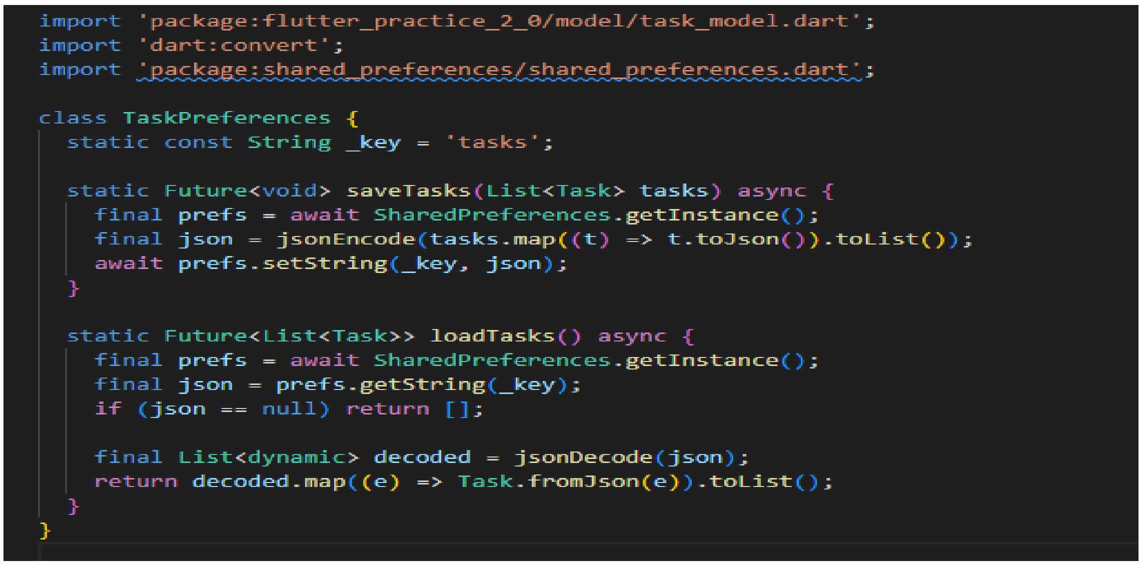
```dart
class Task {
  String title;
  String description;
  bool isCompleted;

  Task({
    required this.title,
    required this.description,
    this.isCompleted = false,
  });

  //Convertir a Map para guardar
  Map<String, dynamic> toJson() => {
        'title': title,
        'description': description,
        'isCompleted': isCompleted,
      };

  factory Task.fromJson(Map<String, dynamic> json) => Task(
        title: json['title'] ?? 'Sin titulo',
        description: json['description'] ?? 'Sin descripción',
        isCompleted: json['isCompleted'] ?? false,
      );
}

```

Para shared preferences, hay que añadir en yaml:



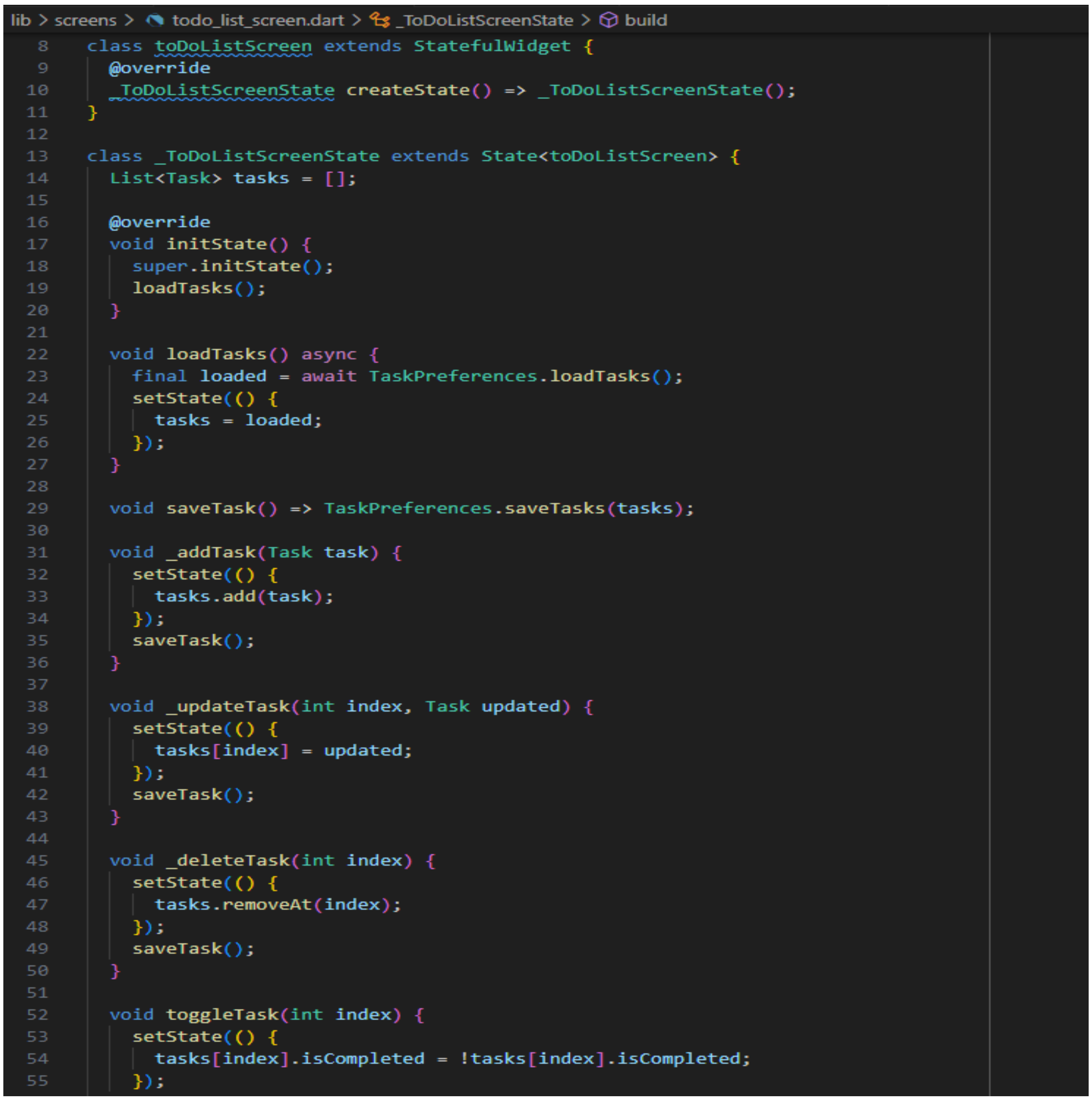
```yaml
dev_dependencies:
  flutter_test:
    sdk: flutter
  shared_preferences: ^2.2.2
```

Shared preferences



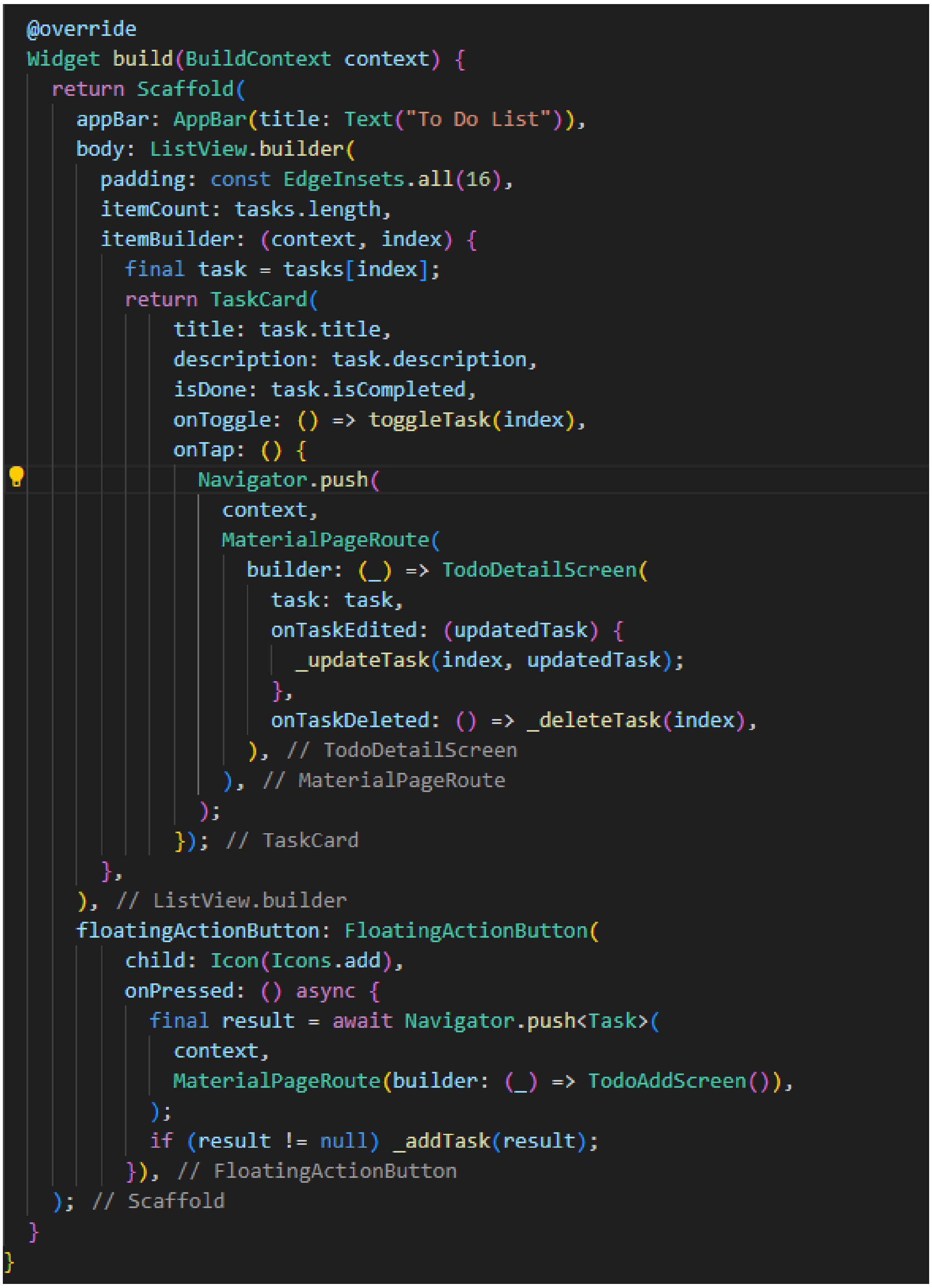
```dart
import 'package:flutter_practice_2_0/model/task_model.dart';
import 'dart:convert';
import 'package:shared_preferences/shared_preferences.dart';

class TaskPreferences {
  static const String _key = 'tasks';

  static Future<void> saveTasks(List<Task> tasks) async {
    final prefs = await SharedPreferences.getInstance();
    final json = jsonEncode(tasks.map((t) => t.toJson()).toList());
    await prefs.setString(_key, json);
  }

  static Future<List<Task>> loadTasks() async {
    final prefs = await SharedPreferences.getInstance();
    final json = prefs.getString(_key);
    if (json == null) return [];

    final List<dynamic> decoded = jsonDecode(json);
    return decoded.map((e) => Task.fromJson(e)).toList();
  }
}
```

Ejemplo de list screen



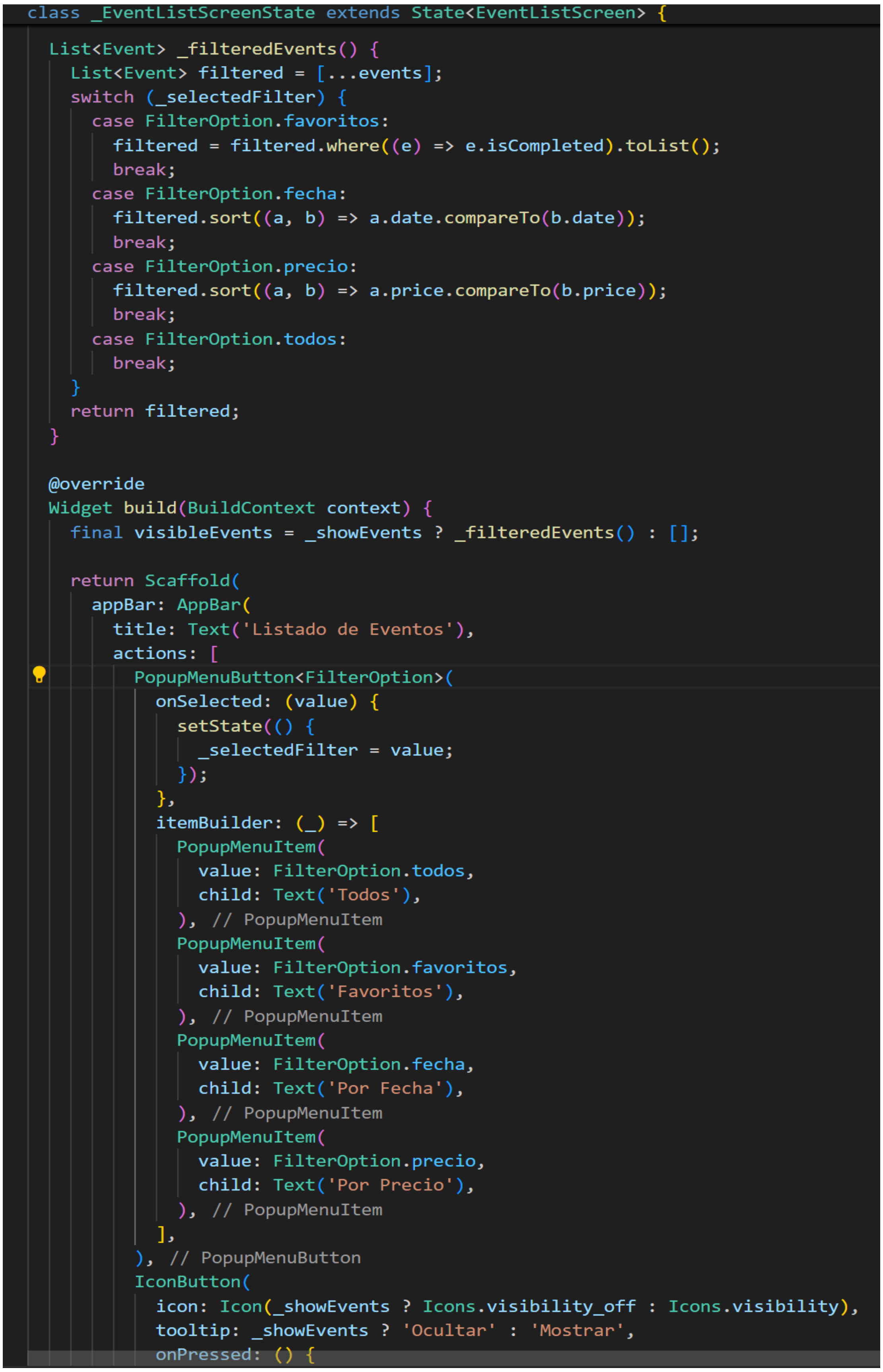
lib > screens > todo_list_screen.dart > _ToDoListScreenState > build

```dart
class toDoListScreen extends StatefulWidget {
  @override
  _ToDoListScreenState createState() => _ToDoListScreenState();
}

class _ToDoListScreenState extends State<toDoListScreen> {
  List<Task> tasks = [];

  @override
  void initState() {
    super.initState();
    loadTasks();
  }

  void loadTasks() async {
    final loaded = await TaskPreferences.loadTasks();
    setState(() {
      tasks = loaded;
    });
  }

  void saveTask() => TaskPreferences.saveTasks(tasks);

  void _addTask(Task task) {
    setState(() {
      tasks.add(task);
    });
    saveTask();
  }

  void _updateTask(int index, Task updated) {
    setState(() {
      tasks[index] = updated;
    });
    saveTask();
  }

  void _deleteTask(int index) {
    setState(() {
      tasks.removeAt(index);
    });
    saveTask();
  }

  void toggleTask(int index) {
    setState(() {
      tasks[index].isCompleted = !tasks[index].isCompleted;
    });
```

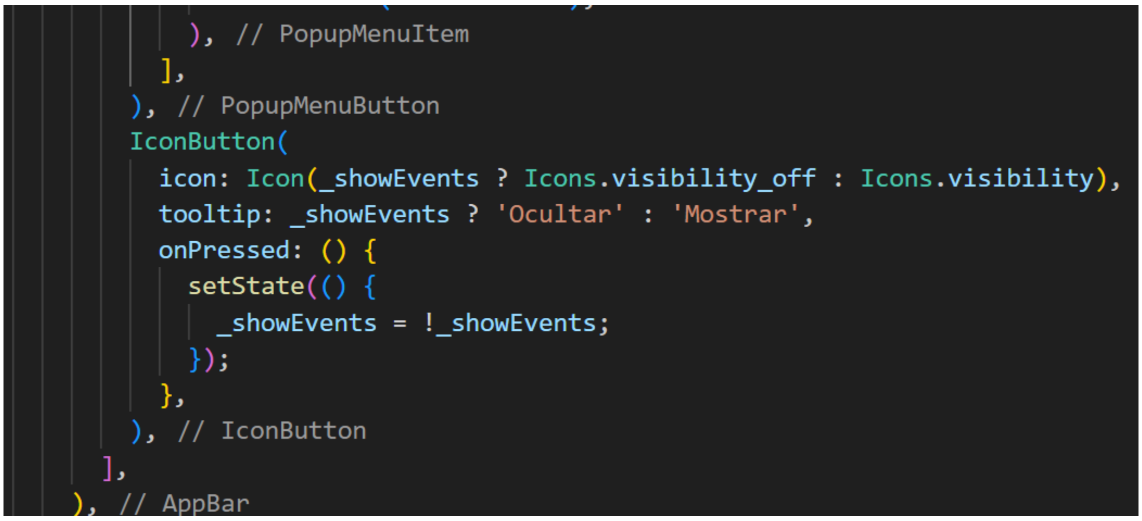


```dart
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("To Do List")),
      body: ListView.builder(
        padding: const EdgeInsets.all(16),
        itemCount: tasks.length,
        itemBuilder: (context, index) {
          final task = tasks[index];
          return TaskCard(
            title: task.title,
            description: task.description,
            isDone: task.isCompleted,
            onToggle: () => toggleTask(index),
            onTap: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (_) => TodoDetailScreen(
                    task: task,
                    onTaskEdited: (updatedTask) {
                      _updateTask(index, updatedTask);
                    },
                    onTaskDeleted: () => _deleteTask(index),
                  ), // TodoDetailScreen
                ), // MaterialPageRoute
              );
            }); // TaskCard
        },
      ), // ListView.builder
      floatingActionButton: FloatingActionButton(
          child: Icon(Icons.add),
          onPressed: () async {
            final result = await Navigator.push<Task>(
              context,
              MaterialPageRoute(builder: (_) => TodoAddScreen()),
            );
            if (result != null) _addTask(result);
          }), // FloatingActionButton
    ); // Scaffold
  }
}
```

Otro ejemplo: para implementar filtros

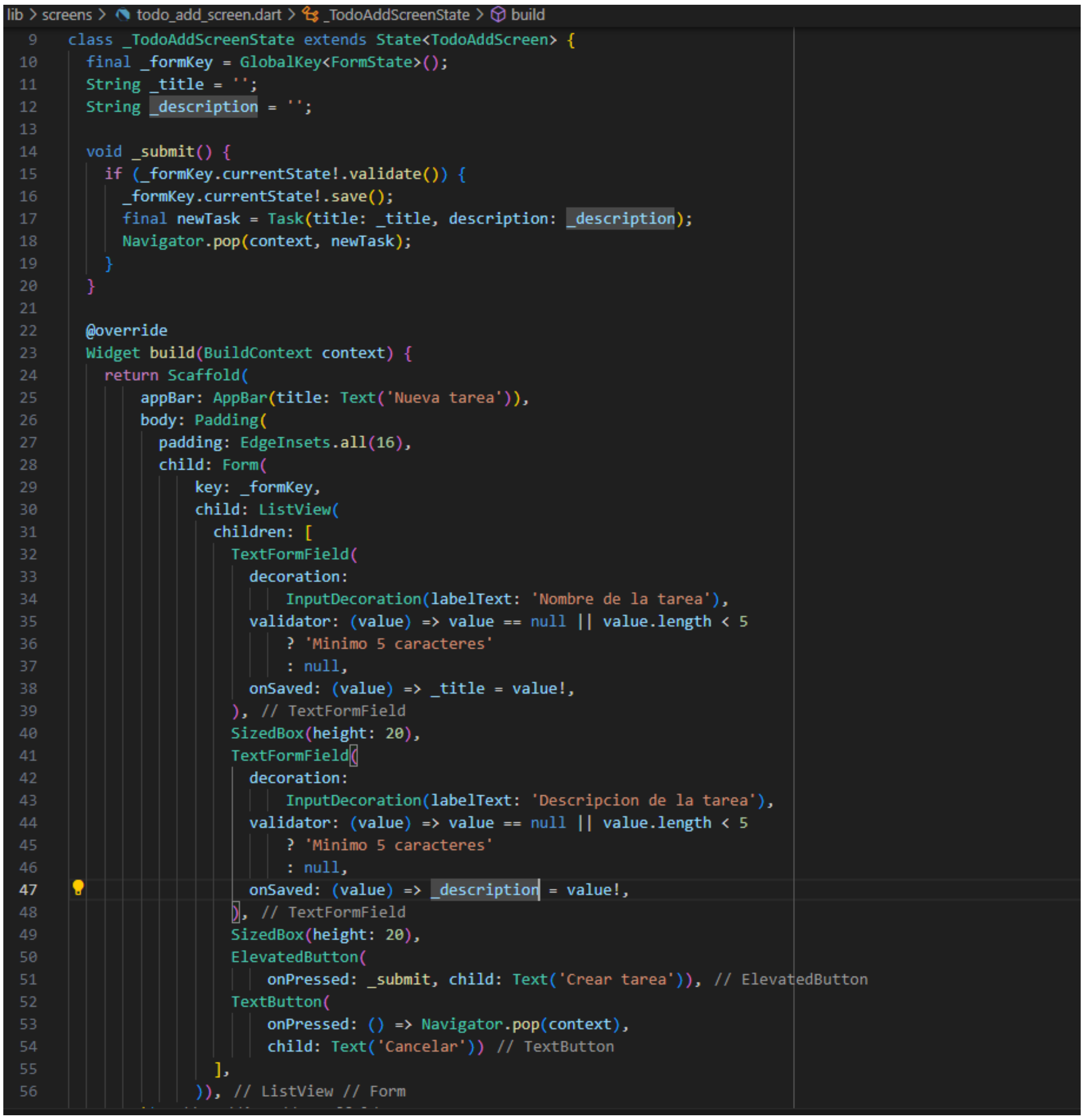


```
class _EventListScreenState extends State<EventListScreen> {

  List<Event> _filteredEvents() {
    List<Event> filtered = [...events];
    switch (_selectedFilter) {
      case FilterOption.favoritos:
        filtered = filtered.where((e) => e.isCompleted).toList();
        break;
      case FilterOption.fecha:
        filtered.sort((a, b) => a.date.compareTo(b.date));
        break;
      case FilterOption.precio:
        filtered.sort((a, b) => a.price.compareTo(b.price));
        break;
      case FilterOption.todos:
        break;
    }
    return filtered;
  }

  @override
  Widget build(BuildContext context) {
    final visibleEvents = _showEvents ? _filteredEvents() : [];

    return Scaffold(
      appBar: AppBar(
        title: Text('Listado de Eventos'),
        actions: [
          PopupMenuButton<FilterOption>(
            onSelected: (value) {
              setState(() {
                _selectedFilter = value;
              });
            },
            itemBuilder: (_) => [
              PopupMenuItem(
                value: FilterOption.todos,
                child: Text('Todos'),
              ), // PopupMenuItem
              PopupMenuItem(
                value: FilterOption.favoritos,
                child: Text('Favoritos'),
              ), // PopupMenuItem
              PopupMenuItem(
                value: FilterOption.fecha,
                child: Text('Por Fecha'),
              ), // PopupMenuItem
              PopupMenuItem(
                value: FilterOption.precio,
                child: Text('Por Precio'),
              ), // PopupMenuItem
            ],
          ), // PopupMenuButton
          IconButton(
            icon: Icon(_showEvents ? Icons.visibility_off : Icons.visibility),
            tooltip: _showEvents ? 'Ocultar' : 'Mostrar',
            onPressed: () {
```



```
            ), // PopupMenuItem
          ],
        ), // PopupMenuButton
        IconButton(
          icon: Icon(_showEvents ? Icons.visibility_off : Icons.visibility),
          tooltip: _showEvents ? 'Ocultar' : 'Mostrar',
          onPressed: () {
            setState(() {
              _showEvents = !_showEvents;
            });
          },
        ), // IconButton
      ],
    ), // AppBar
```

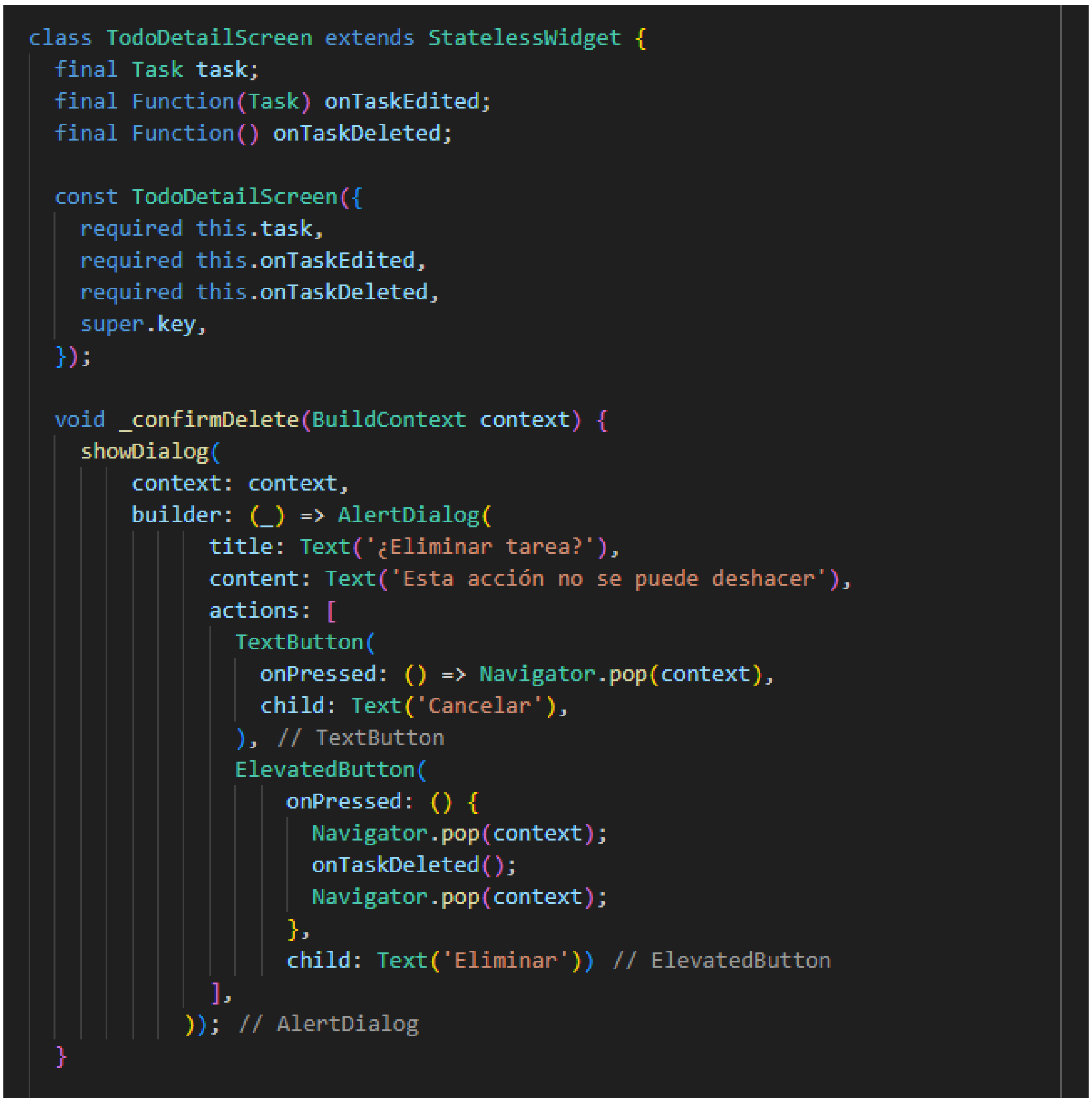
Addscreen



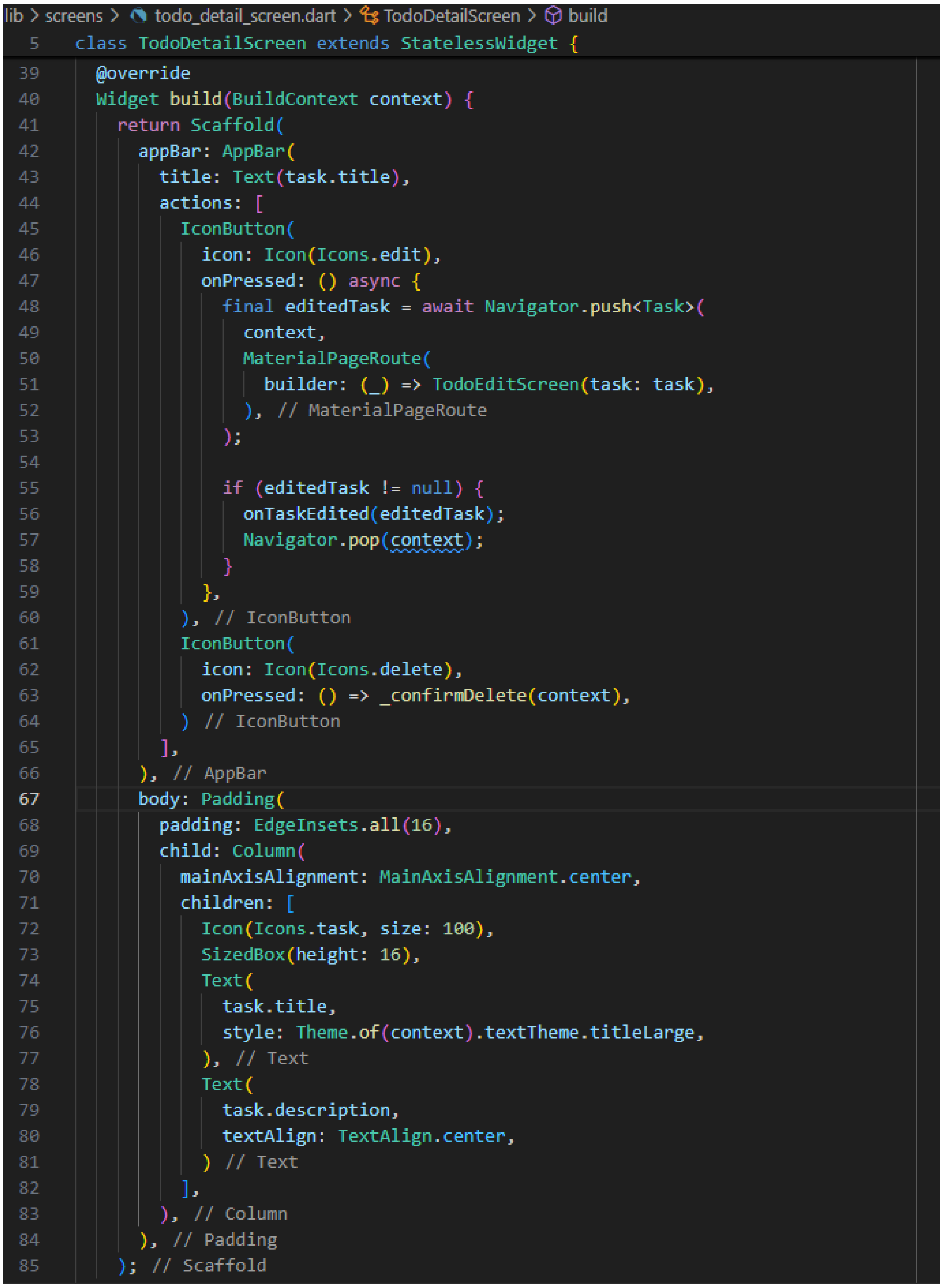
lib > screens > todo_add_screen.dart > _TodoAddScreenState > build

```
class _TodoAddScreenState extends State<TodoAddScreen> {
  final _formKey = GlobalKey<FormState>();
  String _title = '';
  String _description = '';

  void _submit() {
    if (_formKey.currentState!.validate()) {
      _formKey.currentState!.save();
      final newTask = Task(title: _title, description: _description);
      Navigator.pop(context, newTask);
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(title: Text('Nueva tarea')),
        body: Padding(
          padding: EdgeInsets.all(16),
          child: Form(
              key: _formKey,
              child: ListView(
                children: [
                  TextFormField(
                    decoration:
                        InputDecoration(labelText: 'Nombre de la tarea'),
                    validator: (value) => value == null || value.length < 5
                        ? 'Minimo 5 caracteres'
                        : null,
                    onSaved: (value) => _title = value!,
                  ), // TextFormField
                  SizedBox(height: 20),
                  TextFormField(
                    decoration:
                        InputDecoration(labelText: 'Descripcion de la tarea'),
                    validator: (value) => value == null || value.length < 5
                        ? 'Minimo 5 caracteres'
                        : null,
                    onSaved: (value) => _description = value!,
                  ), // TextFormField
                  SizedBox(height: 20),
                  ElevatedButton(
                      onPressed: _submit, child: Text('Crear tarea')), // ElevatedButton
                  TextButton(
                      onPressed: () => Navigator.pop(context),
                      child: Text('Cancelar')) // TextButton
                ],
              )), // ListView // Form
```

DetailScreen



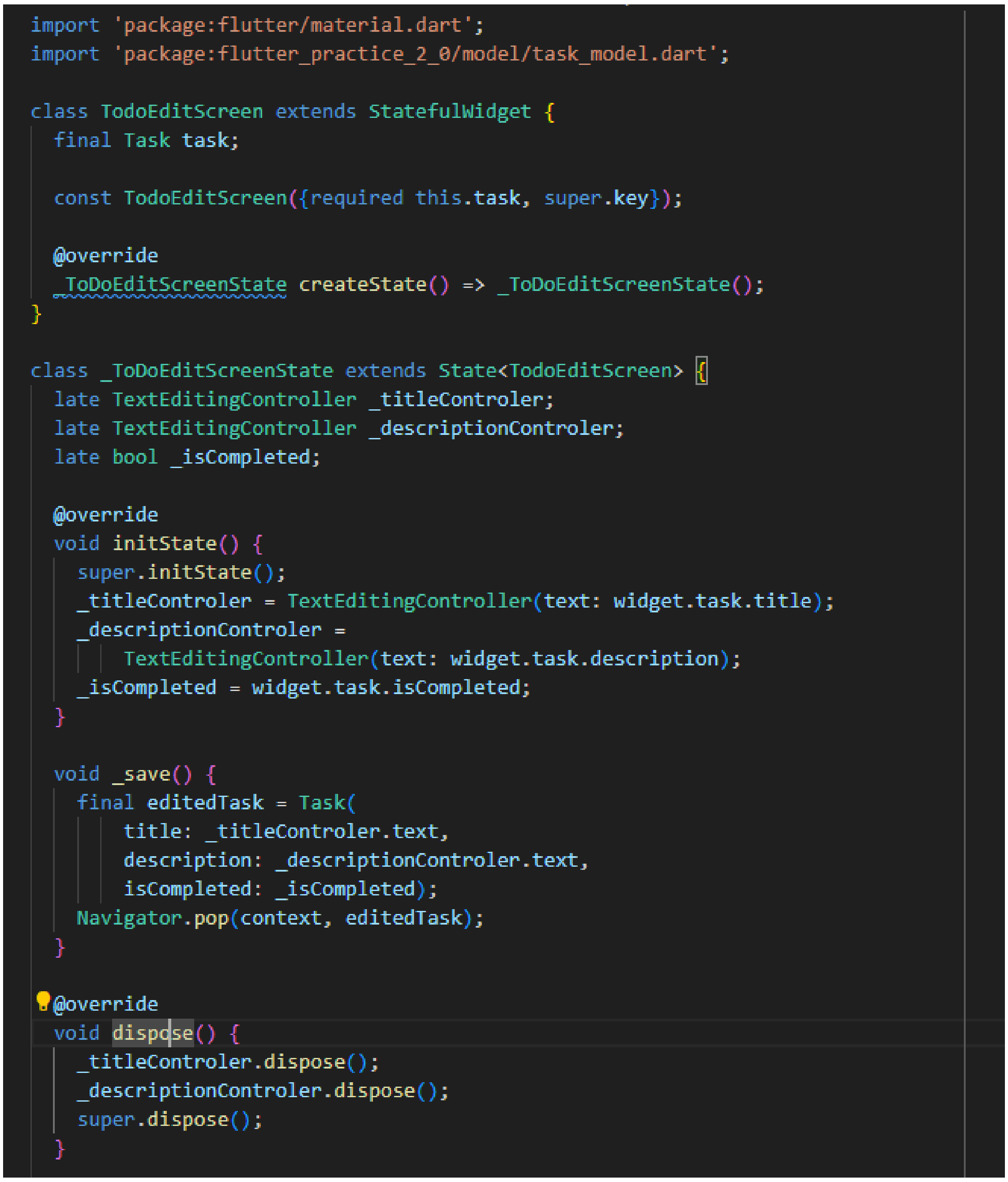
```dart
class TodoDetailScreen extends StatelessWidget {
  final Task task;
  final Function(Task) onTaskEdited;
  final Function() onTaskDeleted;

  const TodoDetailScreen({
    required this.task,
    required this.onTaskEdited,
    required this.onTaskDeleted,
    super.key,
  });

  void _confirmDelete(BuildContext context) {
    showDialog(
        context: context,
        builder: (_) => AlertDialog(
              title: Text('¿Eliminar tarea?'),
              content: Text('Esta acción no se puede deshacer'),
              actions: [
                TextButton(
                  onPressed: () => Navigator.pop(context),
                  child: Text('Cancelar'),
                ), // TextButton
                ElevatedButton(
                    onPressed: () {
                      Navigator.pop(context);
                      onTaskDeleted();
                      Navigator.pop(context);
                    },
                    child: Text('Eliminar')) // ElevatedButton
              ],
            )); // AlertDialog
  }
```



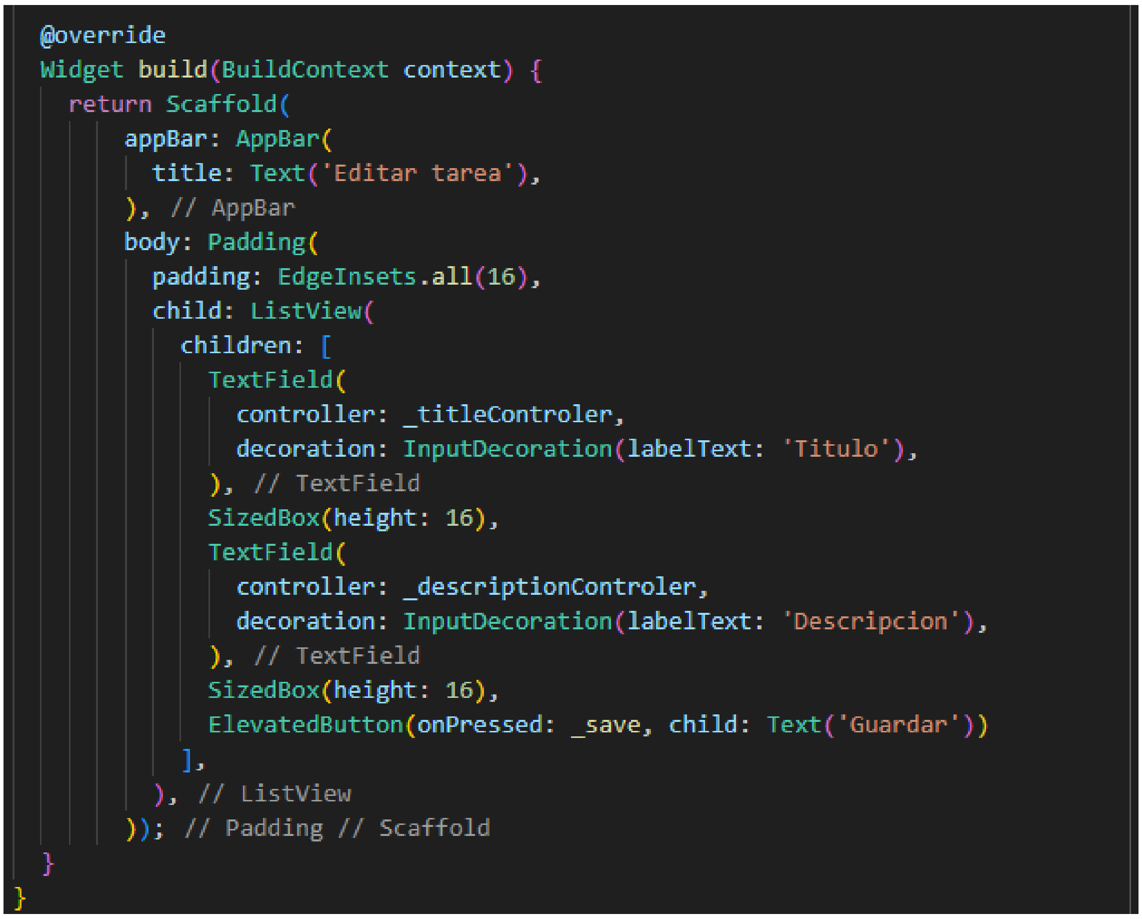
lib > screens > todo_detail_screen.dart > TodoDetailScreen > build

```dart
class TodoDetailScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(task.title),
        actions: [
          IconButton(
            icon: Icon(Icons.edit),
            onPressed: () async {
              final editedTask = await Navigator.push<Task>(
                context,
                MaterialPageRoute(
                  builder: (_) => TodoEditScreen(task: task),
                ), // MaterialPageRoute
              );

              if (editedTask != null) {
                onTaskEdited(editedTask);
                Navigator.pop(context);
              }
            },
          ), // IconButton
          IconButton(
            icon: Icon(Icons.delete),
            onPressed: () => _confirmDelete(context),
          ) // IconButton
        ],
      ), // AppBar
      body: Padding(
        padding: EdgeInsets.all(16),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Icon(Icons.task, size: 100),
            SizedBox(height: 16),
            Text(
              task.title,
              style: Theme.of(context).textTheme.titleLarge,
            ), // Text
            Text(
              task.description,
              textAlign: TextAlign.center,
            ) // Text
          ],
        ), // Column
      ), // Padding
    ); // Scaffold
```

EditScreen



```dart
import 'package:flutter/material.dart';
import 'package:flutter_practice_2_0/model/task_model.dart';

class TodoEditScreen extends StatefulWidget {
  final Task task;

  const TodoEditScreen({required this.task, super.key});

  @override
  _ToDoEditScreenState createState() => _ToDoEditScreenState();
}

class _ToDoEditScreenState extends State<TodoEditScreen> {
  late TextEditingController _titleControler;
  late TextEditingController _descriptionControler;
  late bool _isCompleted;

  @override
  void initState() {
    super.initState();
    _titleControler = TextEditingController(text: widget.task.title);
    _descriptionControler =
        TextEditingController(text: widget.task.description);
    _isCompleted = widget.task.isCompleted;
  }

  void _save() {
    final editedTask = Task(
        title: _titleControler.text,
        description: _descriptionControler.text,
        isCompleted: _isCompleted);
    Navigator.pop(context, editedTask);
  }

  @override
  void dispose() {
    _titleControler.dispose();
    _descriptionControler.dispose();
    super.dispose();
  }
```



```dart
@override
Widget build(BuildContext context) {
  return Scaffold(
      appBar: AppBar(
        title: Text('Editar tarea'),
      ), // AppBar
      body: Padding(
        padding: EdgeInsets.all(16),
        child: ListView(
          children: [
            TextField(
              controller: _titleControler,
              decoration: InputDecoration(labelText: 'Titulo'),
            ), // TextField
            SizedBox(height: 16),
            TextField(
              controller: _descriptionControler,
              decoration: InputDecoration(labelText: 'Descripcion'),
            ), // TextField
            SizedBox(height: 16),
            ElevatedButton(onPressed: _save, child: Text('Guardar'))
          ],
        ), // ListView
      )); // Padding // Scaffold
  }
}
```