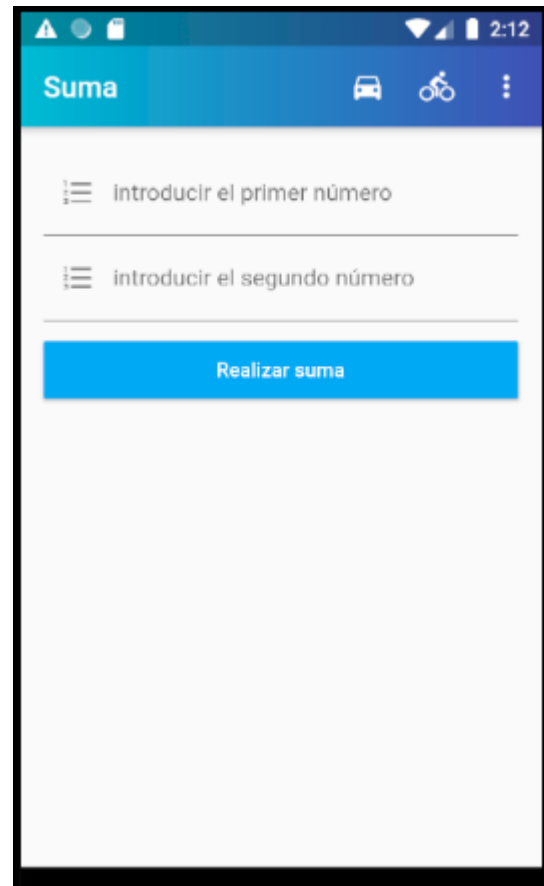
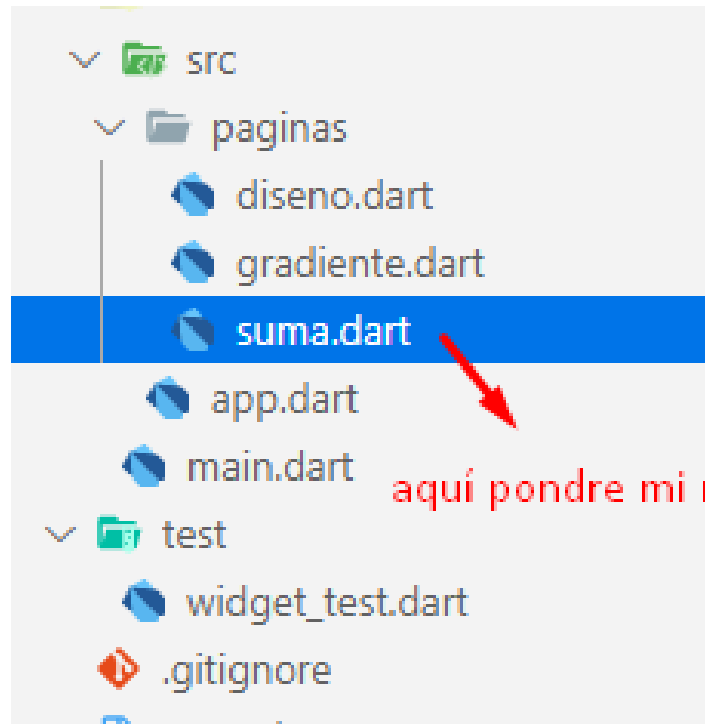


Menú e iconos en la appbar

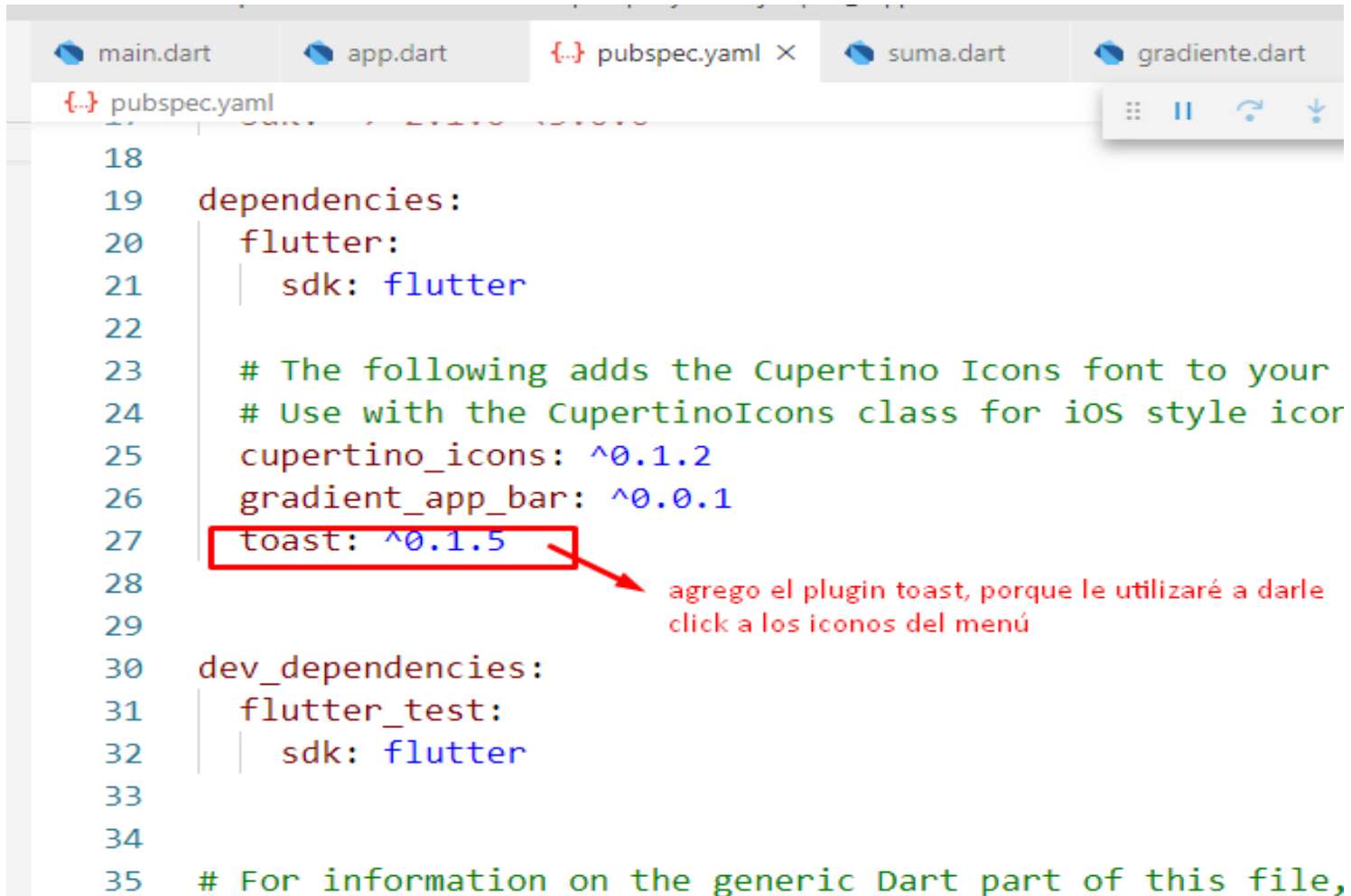


Para esta práctica necesitaremos 3 pantallas diferentes, una que contendrá el menú y las otras que serán llamadas desde el menú



```
8  @override
9  State<Sta
10 }
11
12 // Define l
13 // nuestro
14 class _Suma
15 // Crea u
16 // del Te
17 /*Ya que
```

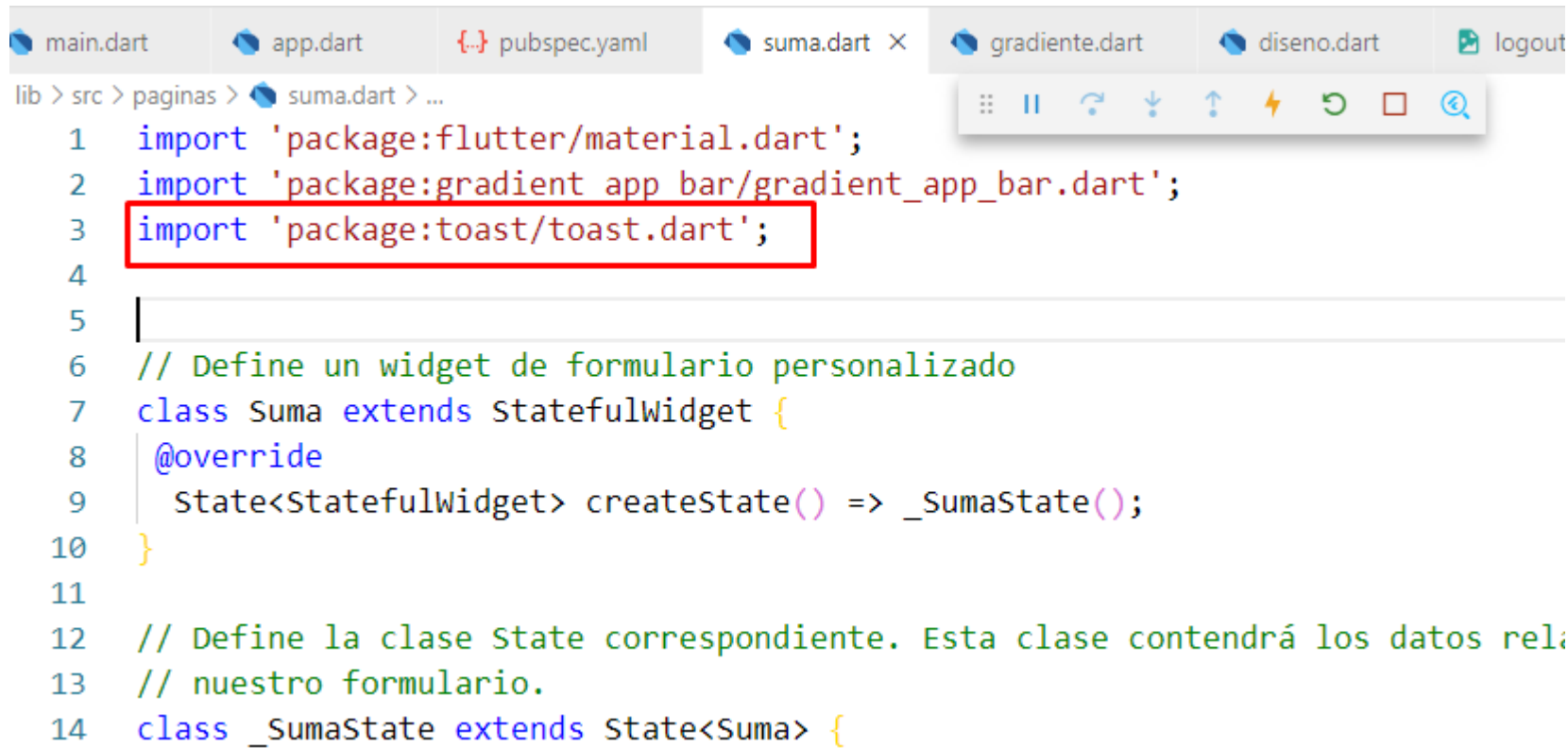
Abrir archivo pubspec.yaml



```
18
19 dependencies:
20   flutter:
21     sdk: flutter
22
23   # The following adds the Cupertino Icons font to your
24   # Use with the CupertinoIcons class for iOS style icons
25   cupertino_icons: ^0.1.2
26   gradient_app_bar: ^0.0.1
27   toast: ^0.1.5
28
29 dev_dependencies:
30   flutter_test:
31     sdk: flutter
32
33
34
35 # For information on the generic Dart part of this file,
```

agrego el plugin toast, porque le utilizaré a darle click a los iconos del menú

En el archivo suma o donde podrán su menú, agregan la llamada al toast



The screenshot shows an IDE with several tabs: main.dart, app.dart, pubspec.yaml, suma.dart (active), gradiente.dart, diseno.dart, and logout. The suma.dart file is open, showing the following code:

```
lib > src > paginas > suma.dart > ...
1 import 'package:flutter/material.dart';
2 import 'package:gradient_app_bar/gradient_app_bar.dart';
3 import 'package:toast/toast.dart';
4
5 |
6 // Define un widget de formulario personalizado
7 class Suma extends StatefulWidget {
8   @override
9   State<StatefulWidget> createState() => _SumaState();
10 }
11
12 // Define la clase State correspondiente. Esta clase contendrá los datos rela
13 // nuestro formulario.
14 class _SumaState extends State<Suma> {
```

The third line, `import 'package:toast/toast.dart';`, is highlighted with a red rectangle. A toolbar with various icons is visible above the code editor.



En el appbar


```
main.dart  app.dart  pubspec.yaml  suma.dart  gradiente.dart  diseno.dart  logoutm.png
lib > src > paginas > suma.dart > _SumaState > build
52  return Scaffold(
53    appBar: GradientAppBar(
54      title: Text('Suma'),
55      backgroundColorStart: Colors.cyan,
56      backgroundColorEnd: Colors.indigo,
57      actions: <Widget>[
58        // action button
59        IconButton(
60          icon: Icon(Icons.directions_car),
61          onPressed: () {
62
63            Toast.show("Pruebitaa", context, duration: Toast.LENGTH_SHORT, gravity: Toast.BOTTOM);
64
65          },
66        ), // IconButton
67        // action button
68        IconButton(
69          icon: Icon( Icons.directions_bike),
70          onPressed: () {
71
72            Toast.show("Pruebita 2", context, duration: Toast.LENGTH_SHORT, gravity: Toast.BOTTOM);
73
74          },
75        ), // IconButton
76      ], // <Widget>[]
77    ), // GradientAppBar
```


cuando presiono el icono enviará el toast con un mensaje

resultado

Suma






 Introducir el primer número


 introducir el segundo número

Realizar suma

Suma



 Introducir el primer número

 introducir el segundo número

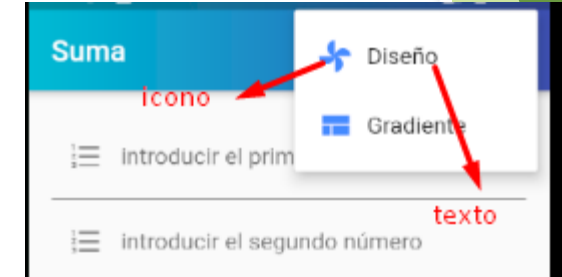
Realizar suma

Pruebitaa

En el mismo archivo donde irá el menú, agrego el siguiente código

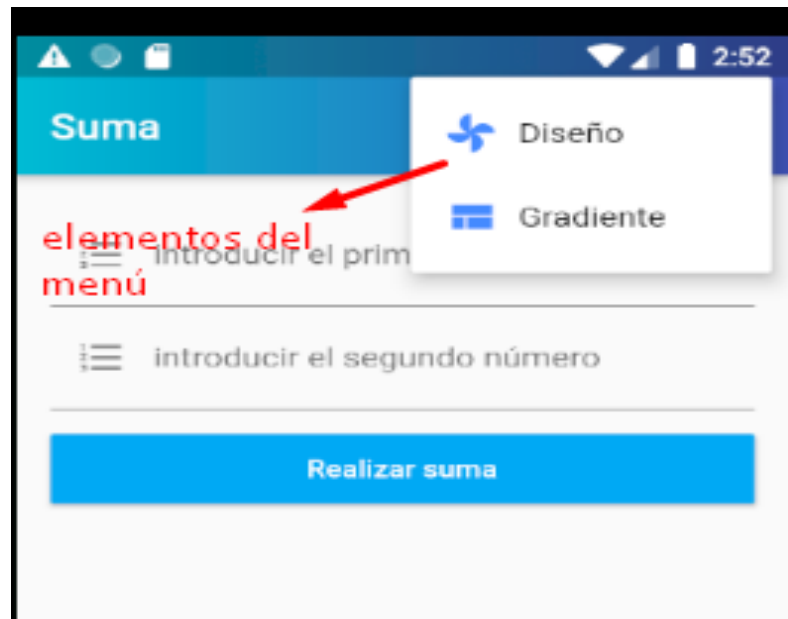
```
main.dart app.dart pubspec.yaml suma.dart • gradiente.dart diseno.dart logoutm.png
lib > src > paginas > suma.dart > _SumaState
152      ), // ListView
153      ), // Padding
154
155    ); // Scaffold
156  }
157
158
159 } //fin de la clase que extiende del state
160
161 class Choice {
162   final String title;
163   final IconData icon;
164
165   const Choice({this.title, this.icon});
166
167 }
168
169
170 const List<Choice> choices = const <Choice>[
171   const Choice(title: 'Diseño', icon: Icons.toys),
172   const Choice(title: 'Gradiente', icon: Icons.view_compact),
173
174 ]; // <Choice>[]
```

sirve para configurar de que esta compuesto cada uno de los elemento del menú, en mi caso cada elemento del menú puse que será un texto y un icono



```
.67 |  
.68 | }  
.69 |  
.70 | const List<Choice> choices = const <Choice>[  
.71 |     const Choice(title: 'Diseño', icon: Icons.toys),  
.72 |     const Choice(title: 'Gradiente', icon: Icons.view_compact),  
.73 |  
.74 | ]; // <Choice>[]
```

esto es la lista de elemento de los
que estará compuesto mi menú cada uno es
un objeto de clase choice



main.dart

app.dart

pubspec.yaml

suma.dart

gradiente.dart

diseno.dart

logoutm.png

lib > src > paginas > suma.dart > _SumaState

```
13 // nuestro formulario.
14 class _SumaState extends State<Suma> {
15   // Crea un controlador de texto. Lo usaremos para recuperar el valor actual
16   // del TextField!
17   /*Ya que el TextField no tiene un ID como en Android, el texto no puede ser recuperado sobre de
18   final myController = TextEditingController();
19   final myControllernum2 = TextEditingController();
20
21   void _select(Choice choice) {
22     // Causes the app to rebuild with the new _selectedChoice.
23     setState(() {
24       switch(choice.title)
25       {
26         case 'Diseño':
27           Navigator.pushNamed(context, '/diseno');
28           break;
29
30         case 'Gradiente':
31           Navigator.pushNamed(context, '/gradiente');
32           break;
33       }
34     });
35   }
36 }
37 }
```

dentro de la clase agrego el siguiente código

opción que enviará al seleccionar una opción
para hacer la llamada a otra pantalla, esto trabaja
con código que pondremos en el archivo app.dart

Función que se ejecuta al
seleccionar alguna opción
del menú

main.dart

app.dart

pubspec.yaml

suma.dart

gradiente.dart

diseno.dart

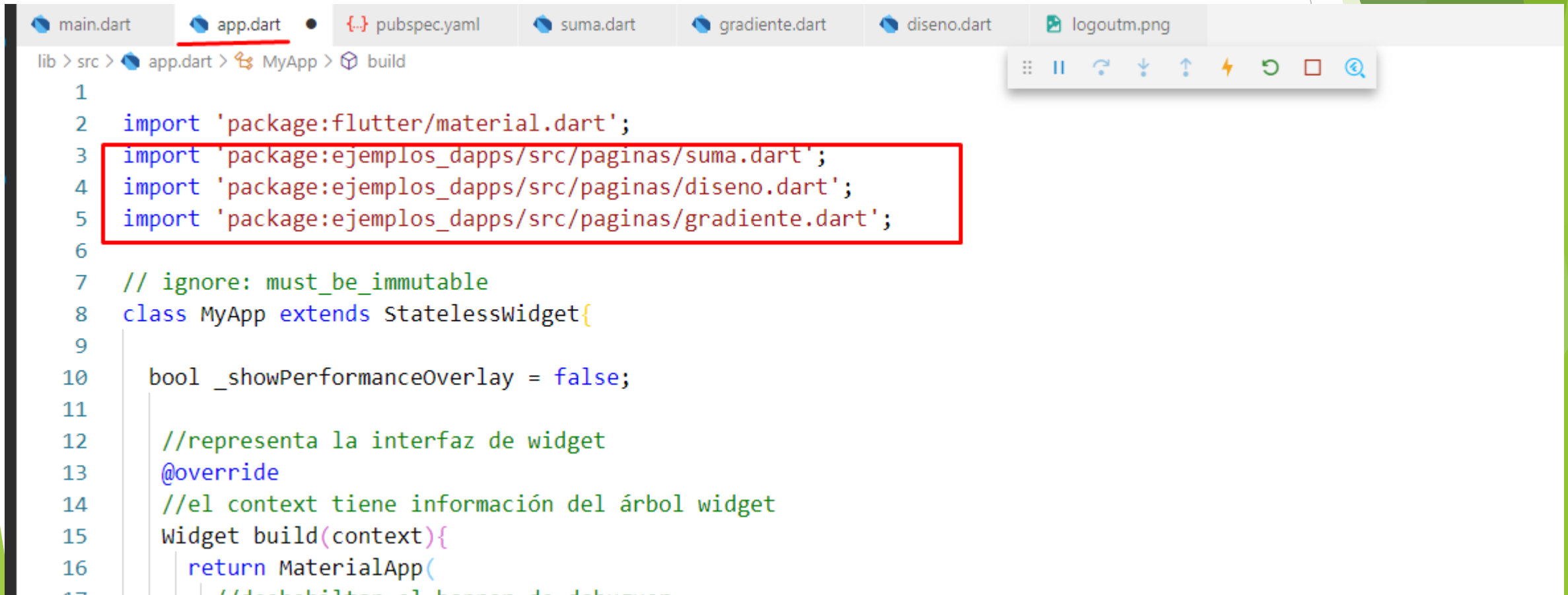
logoutm.png

lib > src > paginas > suma.dart > _SumaState

```
73      },
74    ), // IconButton
75
76    // overflow menu
77    PopupMenuButton<Choice>(
78      //cuando seleccciono una opción llama a la función _select y envia la opción seleccionada
79      onSelect: _select,
80      itemBuilder: (BuildContext context) {
81        return choices.skip(0).map((Choice choice) {
82          return PopupMenuItem<Choice>(
83            value: choice,
84            child: Row(
85              children: <Widget>[
86                Icon(choice.icon, color: Colors.blueAccent),
87                Padding(padding: const EdgeInsets.all(5)),
88                Text(choice.title),
89              ],
90            ), // <Widget>[] // Row
91          ); // PopupMenuItem
92        }).toList();
93      },
94    ), // PopupMenuButton
95
96    ], // <Widget>[]
97  ), // GradientAppBar
```

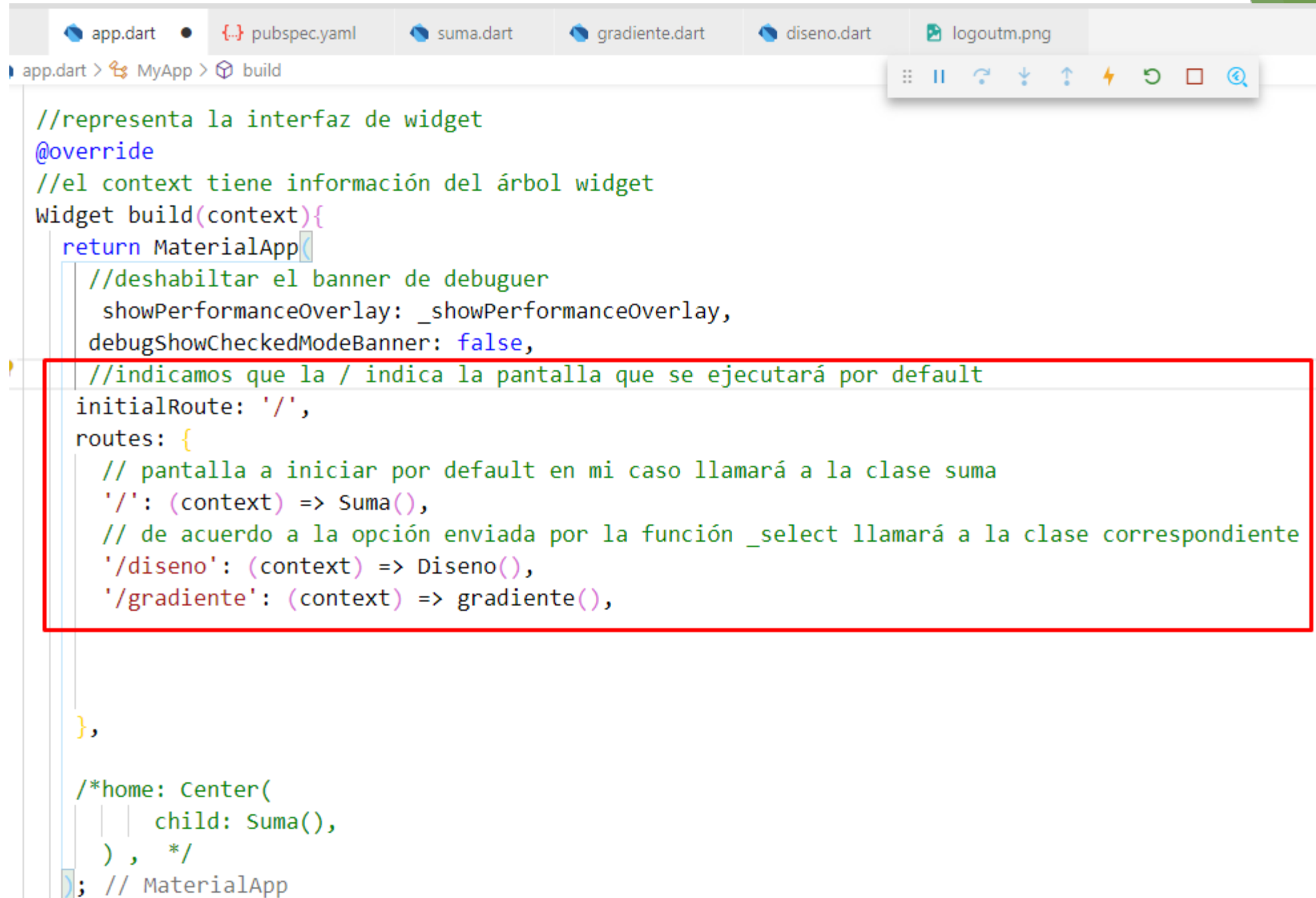
toda las opciones del menú de la lista declarada previamente, recordando que es de una clase que tiene icono y texto

En el archivo app.dart



The screenshot shows an IDE window with several tabs: main.dart, app.dart (selected), pubspec.yaml, suma.dart, gradiente.dart, diseno.dart, and logoutm.png. The app.dart file is open, showing Dart code. Lines 3, 4, and 5, which are import statements for local packages, are highlighted with a red rectangle. The code includes a comment about ignoring the must_be_immutable warning and the start of a StatelessWidget class named MyApp.

```
lib > src > app.dart > MyApp > build
1
2 import 'package:flutter/material.dart';
3 import 'package:ejemplos_dapps/src/paginas/suma.dart';
4 import 'package:ejemplos_dapps/src/paginas/disenio.dart';
5 import 'package:ejemplos_dapps/src/paginas/gradiente.dart';
6
7 // ignore: must_be_immutable
8 class MyApp extends StatelessWidget{
9
10     bool _showPerformanceOverlay = false;
11
12     //representa la interfaz de widget
13     @override
14     //el context tiene información del árbol widget
15     Widget build(context){
16         return MaterialApp(
17             //definiendo el tema de la aplicación
```



```
app.dart • pubspec.yaml suma.dart gradiente.dart diseno.dart logoutm.png
app.dart > MyApp > build

//representa la interfaz de widget
@override
//el context tiene información del árbol widget
Widget build(context){
  return MaterialApp(
    //deshabilitar el banner de debuguer
    showPerformanceOverlay: _showPerformanceOverlay,
    debugShowCheckedModeBanner: false,
    //indicamos que la / indica la pantalla que se ejecutará por default
    initialRoute: '/',
    routes: {
      // pantalla a iniciar por default en mi caso llamará a la clase suma
      '/': (context) => Suma(),
      // de acuerdo a la opción enviada por la función _select llamará a la clase correspondiente
      '/diseno': (context) => Diseno(),
      '/gradiente': (context) => gradiente(),
    },
    /*home: Center(
      child: Suma(),
    ), */
  ); // MaterialApp
```

Resultado final

Link para regresar a la pantalla principal

