

GUIA FLUTTER #2

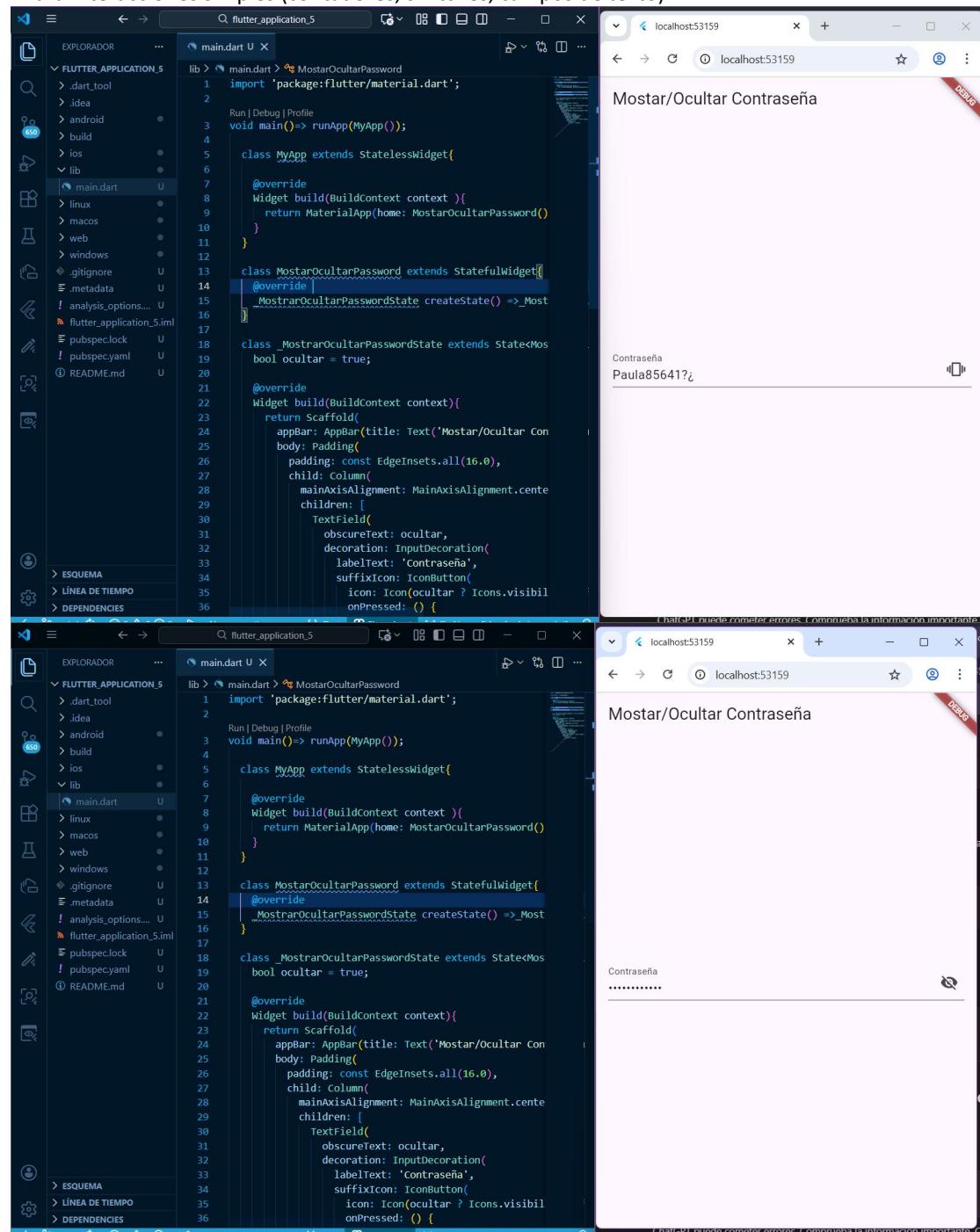
PAULA ANDREA CONDE AGUJA
FICHA. 3197815

Diego Alejandro Boada Morales
Aplicaciones Móviles II

SEDE SENA
CENTRO DE SERVICIOS FINANCIEROS

1. Gestión de estado

- Para interacciones simples (contadores, switches, campos de texto).



The screenshot shows two side-by-side views of a Flutter application running in a browser. The left view is the code editor (Android Studio) displaying the file `main.dart`. The right view is a browser window showing the app's interface.

Code (main.dart):

```
lib > main.dart > MostarOcultarPassword
1 import 'package:flutter/material.dart';
2
3 void main()=> runApp(MyApp());
4
5 class MyApp extends StatelessWidget{
6
7   @override
8   Widget build(BuildContext context ){
9     return MaterialApp(home: MostarOcultarPassword());
10 }
11
12 class MostarOcultarPassword extends StatefulWidget{
13   @override
14   MostarOcultarPasswordState createState() => _MostarOcultarPasswordState();
15 }
16
17 class _MostarOcultarPasswordState extends State<MostarOcultarPassword>{
18   bool ocultar = true;
19
20   @override
21   Widget build(BuildContext context){
22     return Scaffold(
23       appBar: AppBar(title: Text('Mostar/Ocultar Contraseña')),
24       body: Padding(
25         padding: const EdgeInsets.all(16.0),
26         child: Column(
27           mainAxisAlignment: MainAxisAlignment.center,
28           children: [
29             TextField(
30               obscureText: ocultar,
31               decoration: InputDecoration(
32                 labelText: 'Contraseña',
33                 suffixIcon: IconButton(
34                   icon: Icon(ocultar ? Icons.visibility : Icons.visibility_off),
35                   onPressed: () {
36                     setState(() {
37                       ocultar = !ocultar;
38                     });
39                 },
40               ),
41             ),
42           ],
43         ),
44       ),
45     );
46   }
47 }
```

Browser View:

The browser window displays the app titled "Mostar/Ocultar Contraseña". It contains a single text input field labeled "Contraseña". When the code is run, the text "Paula85641?" is visible in the field. A red "DEBUG" indicator is visible in the top right corner of the browser window.

2. Lifting State Up

- Cuando un widget padre necesita compartir el estado con sus hijos.

The screenshot shows the VS Code interface with the main.dart file open in the editor. The code defines a StatelessWidget MyApp that contains a Scaffold with a appBar and a body. The body is a StatelessWidget ListaTareas. The ListaTareasState class maintains a list of Tarea objects and provides a toggleCompletada method to change the completed status of an item. A browser window on the right shows the resulting application with three items: 'Talleres de Flutter', 'Preparar regalos de Navidad', and 'Viajar para Año Nuevo'. The 'Preparar regalos de Navidad' item is checked as completed.

```
lib/main.dart
void main() => runApp(MyApp());
class Tarea {
  String titulo;
  bool completada;
  Tarea(this.titulo, this.completada);
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text('Lista de Tareas')),
        body: ListaTareas(),
      ),
    );
  }
}
class ListaTareas extends StatefulWidget {
  @override
  ListaTareasState createState() => ListaTareasState();
}
class ListaTareasState extends State {
  List<Tarea> tareas = [
    Tarea('Talleres de Flutter', false),
    Tarea('Preparar regalos de Navidad', false),
    Tarea('Viajar para Año Nuevo', false),
  ];
  void toggleCompletada(int index) {
    setState(() {
      tareas[index].completada = !tareas[index].completada;
    });
  }
}
```

3. Providers:

-Provider es una librería oficial recomendada para gestión de estado en Flutter. Es simple, reactiva y escalable.

The screenshot shows the VS Code interface with the lifting.dart file open in the editor. This file demonstrates the Provider package. It defines a ColorModel class with a _color variable and a cambiarColor method. It also defines a StatelessWidget SelectorDeColor that uses a Consumer widget to display a container with a specific color. The browser window on the right shows the application where four colored buttons (yellow, purple, red, black) are displayed, each corresponding to a different color value.

```
lib/main.dart
class colorModel with ChangeNotifier{
  Color _color = Colors.blue;
  Color get color => _color;
  void cambiarColor(Color nuevoColor){
    _color = nuevoColor;
    notifyListeners();
  }
}
class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context){
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('Selector de color')),
        body: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Consumer<ColorModel>(
              builder: (context, colorModel, _) => Container(
                width: 100,
                height: 100,
                color: colorModel.color,
              ),
            ),
            SizedBox(height: 20),
            Wrap(
              spacing: 10,
              children: [
                Colors.yellow, Colors.purple, Colors.red, Colors.black
              ].map(
                (color) => ElevatedButton(
                  onPressed:

```