

Notas sobre Flutter

Fernando Anselmo

v.1.0 em 15 de novembro de 2021

Não contém neste programas completos ou descrições, está no estilo de *HANDS-ON* (mão na massa) composto por anotações soltas para auxiliar no desenvolvimento ou problemas que podem surgir.

1 BÁSICO

Ambiente Dart para testes:

<https://dartpad.dev/>

Novo projeto:

```
$ flutter create -org dev.decus -a java meu-projeto
```

Reconstruir um projeto: Na pasta do projeto

```
$ flutter create .
```

Parar AVD:

```
$ adb shell
```

```
sync && reboot -p
```

Rodar:

```
$ flutter run [-profile] [-release]
```

1.1 Problemas que podem acontecer

Não encontrou **material.dart**:

```
$ flutter doctor -v
```

```
$ flutter packages get
```

```
$ flutter clean
```

Resolução Geral:

```
$ flutter channel dev
```

```
$ flutter doctor
```

```
$ flutter channel master
$ flutter doctor
```

Definir caminhos das variáveis: (os caminhos se referem ao meu SO)

```
$ flutter config --android-sdk="/home/fernando/Android/Sdk"
$ flutter config --android-studio-dir="/opt/android-studio"
$ /home/fernando/Android/Sdk/tools/bin/sdkmanager --install"cmdline-tools;latest"
$ flutter doctor --android-licenses
```

Habilitar ambientes:

```
$ flutter config --enable-linux-desktop
$ flutter config --enable-windows-desktop
$ flutter config --enable-macos-desktop
```

Não use o comando **print**:

log(texto); e adicionar o pacote: import 'dart:developer';

Para implementar classe com a anotação @immutable:

Todas variáveis devem ser **final** e o construtor **const**.

No caso de Cores:

Ao invés de: Colors.grey[300] usar Color(0xFFE0E0E0)

Ao invés de: Colors.grey[100] usar Color(0xFFFF5F5F5)

Para resolver o **accentColor** depreciado:

1º Criar um objeto de ThemeData:

```
1 final ThemeData theme = ThemeData(
2   primarySwatch: white,
3 );
```

2º Usar este objeto no MaterialApp:

```
1 theme: theme.copyWith(
2   colorScheme: theme.colorScheme.copyWith(secondary: Colors.black),
3 ),
```

Tema **Dark**:

```
1 MaterialApp(
2   themeMode: ThemeMode.dark,
3   theme: _lightTheme,
4   darkTheme: _darkTheme,
5   ...
6 ),
```

Aparecer além da barra:

```
1 return Scaffold(
2   extendBodyBehindAppBar: true,
3 );
```

Mudar a barra superior e inferior do Telefone:

```
1 void main() {  
2   var systemUiOverlayStyle = const SystemUiOverlayStyle(  
3     statusBarColor: Colors.orangeAccent, // ou Colors.transparent,  
4     systemNavigationBarColor: Colors.orangeAccent, // ou Colors.transparent,  
5   );  
6   SystemChrome.setSystemUIOverlayStyle(  
7     systemUiOverlayStyle,  
8   );  
9   runApp(const MyApp());  
10 }
```

FlatButton depreciado, trocar para **TextButton** com estilo:

```
1 final ButtonStyle flatButtonStyle = TextButton.styleFrom(  
2   backgroundColor: Colors.grey,  
3   padding: const EdgeInsets.all(0),  
4 );  
5  
6 ...  
7 TextButton(  
8   style: flatButtonStyle,  
9   ...
```

Erro no http:

Criar uma: var url = Uri.parse(END_URL);

E usar esta no HTTP: final response = await http.get(url);

Ou mesmo usar uma url mais completa:

```
1 Uri apiUri = Uri.https('api.tvmaze.com', 'singlesearch/shows',  
2   {'q': 'house', 'embed': 'episodes'});
```

Definir um elemento completo de Cor:

```
1 const MaterialColor white = MaterialColor(  
2   0xFFFFFFFF,  
3   <int, Color>{  
4     50: Color(0xFFFFFFFF),  
5     100: Color(0xFFFFFFFF),  
6     200: Color(0xFFFFFFFF),  
7     300: Color(0xFFFFFFFF),  
8     400: Color(0xFFFFFFFF),  
9     500: Color(0xFFFFFFFF),  
10    600: Color(0xFFFFFFFF),  
11    700: Color(0xFFFFFFFF),  
12    800: Color(0xFFFFFFFF),  
13    900: Color(0xFFFFFFFF),  
14  },  
15 );
```

2 WIDGETS

2.1 Contêineres

Com *children*:

- `Column(children: [])`
- `Row(children: [])`
- `Wrap(children: [])` - Quebra a linha
- `Stack(children: [_image, _text,],)` - Imagem mesclada com texto

Com *child*:

- `Container(child:)`
- `Card(child:)`
- `Align(alignment: Alignment.[tipo], child:)`
- `Padding(padding: const EdgeInsets.all(5.0), child:)`
- `SafeArea(child:)`
- `ClipRRect(borderRadius: BorderRadius.circular(25), child:)`
- `Center(child:)` - Centraliza o texto
- `FittedBox(child:)` - Maximiza o texto
- `Visibility(visible: true/false, child:)` - faz o objeto aparecer ou desaparecer
- `Positioned(top: , left: , height: , width: , child:)`

Decorar um *Container*:

```
1 Container(  
2   decoration: BoxDecoration(  
3     borderRadius: BorderRadius.circular(8.0),  
4     color: Colors.blueAccent,  
5   ),  
6 ),
```

2.2 Divisores

Obter um determinado espaçamento:

- `Expanded(flex: [fator], child:)`
- `Flexible(flex: [fator], child:)`
- `SizedBox(height: alt, width: larg, child:)`
- `Placeholder(fallbackHeight: 200, color: Colors.blue, strokeWidth: 5)`

2.3 Relativos a Imagem

Imagens podem vir de Asset ou Web:

Asset: `Image.asset('assets/images/notfound.png')`

Web: `Image.network('endereço', fit: BoxFit.cover)`

Não quebrar uma imagem vinda da Web:

```
1 Hero(  
2   tag: 'hero',  
3   child: Image.network('https://food.bolt.eu/og-img.jpg'),  
4 ),
```

Criar uma imagem circular:

```
1 CircleAvatar(  
2   radius: ,  
3   backgroundColor: ,  
4   child: _image,  
5 )
```

Exata proporção de uma imagem:

```
1 AspectRatio(  
2   aspectRatio: 1 / 2, (ou 3/4, 7/2)  
3   child: _image,  
4 ),
```

Mudar a opacidade de uma imagem:

```
1 Opacity(  
2   child: _image,  
3   opacity: 0.3,  
4 ),
```

Adicionar um filtro de cor:

```
1 ColorFiltered(  
2   colorFilter: const ColorFilter.mode(  
3     Colors.red,  
4     BlendMode.modulate,  
5   ),  
6   child: _image,  
7 ),
```

Efeito de aparecer (fade) em imagens da Internet:

```
1 FadeInImage.assetNetwork(  
2   placeholder: 'images/loading.gif',  
3   fadeOutDuration: Duration(seconds: 2),  
4   image: 'https://picsum.photos/250?image=9',  
5 ),
```

2.4 Relativos a Textos (Labels)

Texto padrão com estilo:

```

1 Text(
2   episodio['name'] ?? '',
3   style: TextStyle(color: corEpisodio, fontSize: 20),
4 ),

```

Texto selecionável:

```

1 SelectableText('', style: [estilo])

```

Texto com elementos de estilo interno:

```

1 child: RichText(
2   text: TextSpan(
3     style: TextStyle(fontSize: 20, color: Colors.black),
4     children: <TextSpan>[
5       const TextSpan(text: 'Este é um exemplo de '),
6       TextSpan(
7         text: 'RichText',
8         style: TextStyle(
9           fontWeight: FontWeight.bold,
10          color: Colors.blue[400],
11        ),
12      ),
13    ],
14  ),
15 ),

```

2.5 Relativos as Listas de Widgets

Barra de rolagem:

```

1 Scrollbar(
2   isAlwaysShown: true,
3   child: ...
4 ),

```

Slider - Intervalo de Valores:

```

1 Slider(
2   value: valorCorrente,
3   onChanged: (novoValor) {
4     setState(() {
5       valorCorrente = novoValor;
6     });
7   },
8   min: 0,
9   max: 100,
10 ),

```

Chip - Lista de algo

```

1 Chip(
2   avatar: CircleAvatar(
3     child: Text(emails[index].substring(0, 1)),
4   ),
5   label: Text(emails[index]),
6   onDelete: () {
7     setState(() {
8       emails.removeAt(index);
9     });
10  },
11 ),

```

PageView - Várias Visões:

```

1 PageView(
2   scrollDirection: Axis.vertical,
3   children: _widgets,
4 ),

```

Tabela:

```

1 DataTable(
2   columns: [
3     DataColumn(label: nomcol1),
4     DataColumn(label: nomcol2),
5   ],
6   rows: [
7     DataRows(cells: [
8       DataCell(Text(val1c1)),
9       DataCell(Text(val1c2)),
10    ]),
11    DataRows(cells: [
12      DataCell(Text(val2c1)),
13      DataCell(Text(val2c2)),
14    ]),
15  ],
16 )

```

Lista lateral:

```

1 return Scaffold(
2   ...
3   drawer: Drawer(
4     child: _listView,
5   ),
6 );

```

2.6 Diversos e úteis

Dica rápida:

```

1 Tooltip(
2   message: 'Aqui a dica',
3   child: [widget]
4 ),

```

Future.delayed - Espera um tempo:

```

1 onPressed: () async {
2   await Future.delayed(
3     Duration(seconds: 1),
4   );
5   ...
6 }

```

Quando tiver uma espera:

```

1 import 'dart:io'; // import
2 import 'package:flutter/cupertino.dart'; // Cupertino
3
4 Platform.isAndroid
5   ? CircularProgressIndicator(): CupertinoActivityIndicator()

```

Mensagem SnackBar:

```

1 child: Builder(
2   builder: (context) => GestureDetector(
3     onTap: () {
4       ScaffoldMessenger.of(context).showSnackBar(
5         const SnackBar(
6           duration: Duration(seconds: 1),
7           content: Text('This is the Snackbar'),
8         ),
9       );
10    },
11  ),
12 ),

```

Criar um QRCode:

```

1 QrImage(
2   data: 'o que quiser aqui',
3   version: QrVersions.auto,
4   size: 200.0,
5 ),

```

3 TOP PACOTES E PLUGINS

Página dos Pacotes:

<https://pub.dev/>


```
https://pub.dev/packages/introduction_screen
$ flutter pub add introduction_screen
```

```
https://pub.dev/packages/flutter_native_splash
$ flutter pub add flutter_native_splash
```

```
https://pub.dev/packages/google_fonts
$ flutter pub add google_fonts
```

Ex:

```
1 ttfamilyText('Algo', style: GoogleFonts.aguafinaScript().copyWith(fontSize: 40),),
```

```
https://pub.dev/packages/flutter_launch_icons
$ flutter pub add flutter_launch_icons
```

```
https://pub.dev/packages/fluttertoast
$ flutter pub add fluttertoast
```

Ex:

```
1 onPressed: () {
2   Fluttertoast.showToast(
3     msg: 'Mensagem da Torrada',
4     backgroundColor: Colors.deepOrange,
5     textColor: Colors.white,
6     fontSize: 16.0,
7   );
8 },
```

```
https://pub.dev/packages/screenshot
$ flutter pub add screenshot
```

3.1 Plugins para o VS Code

- Dart - Dart Code
- Flutter - Dart Code
- Material Icon Theme - Phillipp Kief
- Pubspec Assist - Jeroen Meijer
- Android Emulator Launcher - Dannark
- Awesome Flutter Snippets - Neevash Ramdial
- Error Lens - Alexander

4 DICAS DE DART

Ao invés de usar: `episodio['name'] == null ? '' : episodio['name']`

Converter para: `episodio['name'] ?? ''`

Obter o tamanho da tela:

```
1 Size size = MediaQuery.of(context).size;  
2 double height = size.height;  
3 double width = size.width;
```

Ao invés de usar: `String text = '\$100';`

Converter para: `String text = r'$100';`