CENTRO UNIVERSITÁRIO IESB

Notas sobre Flutter

Fernando Anselmo

v.1.0 em 12 de dezembro 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(0xFFF5F5F5)`

Para resolver o **accentColor** depreciado:

1º Criar um objeto de ThemeData:

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

2º Usar este objeto no MaterialApp:

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

Tema **Dark**:

```
MaterialApp(
 themeMode: ThemeMode.dark,
 theme: _lightTheme,
 darkTheme: _darkTheme,
 ...
),
```

Aparecer além da barra:

```
return Scaffold(
 extendBodyBehindAppBar: true,
);
```

Mudar a barra superior e inferior do Telefone:

```
void main() {
 var systemUiOverlayStyle = const SystemUiOverlayStyle(
  statusBarColor: Colors.orangeAccent, // ou Colors.transparent,
  systemNavigationBarColor: Colors.orangeAccent, // ou Colors.transparent,
 );
 SystemChrome.setSystemUIOverlayStyle(
  systemUiOverlayStyle,
 );
 runApp(const MyApp());
}
```

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

```
final ButtonStyle flatButtonStyle = TextButton.styleFrom(
 backgroundColor: Colors.grey,
 padding: const EdgeInsets.all(0),
);

...
  TextButton(
   style: flatButtonStyle,
  ...
```

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:

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

Definir um elemento completo de Cor:

```
const MaterialColor white = MaterialColor(
  0xFFFFFFFF,
  <int, Color>{
    50: Color(0xFFFFFFFF),
    100: Color(0xFFFFFFFF),
    200: Color(0xFFFFFFFF),
    300: Color(0xFFFFFFFF),
    400: Color(0xFFFFFFFF),
    500: Color(0xFFFFFFFF),
    600: Color(0xFFFFFFFF),
    700: Color(0xFFFFFFFF),
    800: Color(0xFFFFFFFF),
    900: Color(0xFFFFFFFF),
  },
);
```

2 WIDGETS

2.1 Contâineres

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*:

```
Container(
 decoration: BoxDecoration(
  borderRadius: BorderRadius.circular(8.0),
  color: Colors.blueAccent,
 ),
),
```

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:

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

Criar uma imagem circular:

```
CircleAvatar(
 radius: ,
 backgroundColor: ,
 child: _image,
)
```

Exata proporção de uma imagem:

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

Mudar a opacidade de uma imagem:

```
Opacity(
 child: _image,
 opacity: 0.3,
),
```

Adicionar um filtro de cor:

```
ColorFiltered(
 colorFilter: const ColorFilter.mode(
  Colors.red,
  BlendMode.modulate,
 ),
 child: _image,
),
```

Efeito de aparecer (fade) em imagens da Internet:

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

2.4 Relativos a Textos (Labels)

Texto padrão com estilo:

```
Text(
 episodio['name'] ?? '',
 style: TextStyle(color: corEpisodio, fontSize: 20),
),
```

Texto selecionável:

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

Texto com elementos de estilo interno:

```
child: RichText(
 text: TextSpan(
  style: TextStyle(fontSize: 20, color: Colors.black),
  children: <TextSpan>[
   const TextSpan(text: 'Este é um exemplo de '),
   TextSpan(
    text: 'RichText',
    style: TextStyle(
     fontWeight: FontWeight.bold,
     color: Colors.blue[400],
    ),
   ),
  ],
 ),
),
```

2.5 Relativos as Listas de Widgets

Barra de rolagem:

```
Scrollbar(
 isAlwaysShown: true,
 child: ...
),
```

Slider - Intervalo de Valores:

```
Slider(
 value: valorCorrente,
 onChanged: (novoValor) {
  setState(() {
   valorCorrente = novoValor;
  });
 },
 min: 0,
 max: 100,
),
```

Chip - Lista de algo

```
Chip(
 avatar: CircleAvatar(
  child: Text(emails[index].substring(0, 1)),
 ),
 label: Text(emails[index]),
 onDeleted: () {
  setState(() {
   emails.removeAt(index);
  });
 },
),
```

PageView - Várias Visões:

```
PageView(
 scrollDirection: Axis.vertical,
 children: _widgets,
),
```

Tabela:

```
DataTable(
 columns: [
  DataColumns(label: nomcol1),
  DataColumns(label: nomcol2),
 ],
 rows: [
  DataRows(cells: [
   DataCell(Text(val1c1)),
   DataCell(Text(val1c2)),
  ]),
  DataRows(cells: [
   DataCell(Text(val2c1)),
   DataCell(Text(val2c2)),
  ]),
 ],
)
```

Lista lateral:

```
return Scaffold(
 ...
 drawer: Drawer(
  child: _listView,
 ),
);
```

2.6 Diversos e úteis

Dica rápida:

```
Tooltip(
 message: 'Aqui a dica',
 child: [widget]
),
```

Future.delayed - Espera um tempo:

```
onPressed: () async {
 await Future.delayed(const Duration(seconds: 1));
 ...
}
```

FutureBuilder - Widget:

1º) Construir o método que vai dar a resposta:

```
Future<String> getData() async {
  await Future.delayed(const Duration(seconds: 1));
  // throw "Para testar o erro";
  return "Funciona...";
}
```

2º) Ação no Scaffold:

```
return Scaffold(
   ...
   body: Center(
    child: FutureBuilder(
     future: getData(),
     builder: (context, snapshot) {
      if (snapshot.connectionState == ConnectionState.waiting) {
       return const CircularProgressIndicator.adaptive();
      }
      if (snapshot.hasError) {
       return Text(snapshot.error);
      } else {
       return Column(
        mainAxisSize: MainAxisSize.min,
        children: [
         Text(snapshot.data.toString()),
         ElevatedButton(
          onPressed: () {
           setState(() {});
          },
          child: const Text("Refresh"),
         ),
        ],
       );
      }
     }
    ),
   ),
);
```

Quando tiver uma espera:

```
import 'dart:io'; // import
import 'package:flutter/cupertino.dart'; // Cupertino

Platform.isAndroid
 ? CircularProgressIndicator(): CupertinoActivityIndicator()
```

Mensagem SnackBar:

```
child: Builder(
 builder: (context) => GestureDetector(
  onTap: () {
   ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(
     duration: Duration(seconds: 1),
     content: Text('This is the Snackbar'),
    ),
   );
  },
 ),
),
```

Criar um QrCode:

```
QrImage(
 data: 'o que quiser aqui',
 version: QrVersions.auto,
 size: 200.0,
),
```

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:

```
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:

```
onPressed: () {
 Fluttertoast.showToast(
  msg: 'Mensagem da Torrada',
  backgroundColor: Colors.deepOrange,
  textColor: Colors.white,
  fontSize: 16.0,
 );
},
```

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:

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

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