

Nomes: Gabriel Rodrigues Gietzel e Kayã Costa

O primeiro Widget que me deixou curioso foi esse:

// Flutter code sample for FloatingActionButton

// This example shows how to display a [FloatingActionButton] in a

// [Scaffold], with a pink [backgroundColor] and a thumbs up [Icon].

// ![](https://flutter.github.io/assets-for-api-docs/assets/material/floating\_action\_button.png)

import 'package:flutter/material.dart';

void main() => runApp(MyApp());

/// This Widget is the main application widget.

class MyApp extends StatelessWidget {

static const String \_title = 'Flutter Code Sample';

@override

Widget build(BuildContext context) {

return MaterialApp(

title: \_title,

home: MyStatelessWidget(),

);

}

}

/// This is the stateless widget that the main application instantiates.

class MyStatelessWidget extends StatelessWidget {

MyStatelessWidget({Key key}) : super(key: key);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Floating Action Button'),

),

body: Center(child: const Text('Press the button below!')),

floatingActionButton: FloatingActionButton(

onPressed: () {

// Adicione o código do botão verde aqui!

},

child: Icon(Icons.navigation),

backgroundColor: Colors.green,

),

Uma imagem contendo pássaro, flor

Descrição gerada automaticamente );

}

}

O resultado dele é esse:

É semelhante à um botão de enviar mensagem, eu já vi um parecido em outros apps mas com funções diferentes. Este botão é configurado em “onPressed:”.

Agora um exemplo que mostra como aparecem a maioria das mensagens do sistema:

Future<**void**> \_showMyDialog() **async** {

**return** showDialog<**void**>(

context: context,

barrierDismissible: **false**, ***// user must tap button!***

builder: (BuildContext context) {

**return** AlertDialog(

title: Text('AlertDialog Title'),

content: SingleChildScrollView(

child: ListBody(

children: <Widget>[

Text('This is a demo alert dialog.'),

Text('Would you like to approve of this message?'),

],

),

),

actions: <Widget>[

FlatButton(

child: Text('Approve'),

onPressed: () {

Navigator.of(context).pop();

},

),

],

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente );

},

);

}

O resultado do código é esse:

“user must tap button”, se você clicar em qualquer lugar além do botão nada acontece, geralmente servem para informar algum problema de algum aplicativo.

“Chrome não está respondendo”

Imagem em preto e branco

Descrição gerada automaticamente

Para o último exemplo eu pensei em pegar um bem conhecido, a barra de carregamento em formato circular, mas o código não é tão simples assim, são ao todo 679 linhas, claro que contém texto explicativo também mas é realmente muita coisa.

Até o pequeno menu para alterar o volume tem muita coisa, eu nunca imaginei isso.

Achei um bem simples que pode ser utilizado para formulários:

// Flutter code sample for Radio

// Here is an example of Radio widgets wrapped in ListTiles, which is similar

// to what you could get with the RadioListTile widget.

import 'package:flutter/material.dart';

void main() => runApp(MyApp());

/// This Widget is the main application widget.

class MyApp extends StatelessWidget {

static const String \_title = 'Flutter Code Sample';

@override

Widget build(BuildContext context) {

return MaterialApp(

title: \_title,

home: Scaffold(

appBar: AppBar(title: const Text(\_title)),

body: Center(

child: MyStatefulWidget(),

),

),

);

}

}

enum SingingCharacter { lafayette, jefferson }

class MyStatefulWidget extends StatefulWidget {

MyStatefulWidget({Key key}) : super(key: key);

@override

\_MyStatefulWidgetState createState() => \_MyStatefulWidgetState();

}

class \_MyStatefulWidgetState extends State<MyStatefulWidget> {

SingingCharacter \_character = SingingCharacter.lafayette;

Widget build(BuildContext context) {

return Column(

children: <Widget>[

ListTile(

title: const Text('Lafayette'),

leading: Radio(

value: SingingCharacter.lafayette,

groupValue: \_character,

onChanged: (SingingCharacter value) {

setState(() {

\_character = value;

});

},

),

),

ListTile(

title: const Text('Thomas Jefferson'),

leading: Radio(

value: SingingCharacter.jefferson,

groupValue: \_character,

onChanged: (SingingCharacter value) {

setState(() {

\_character = value;

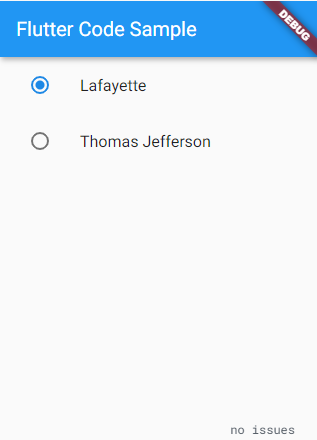
});

},

),

),

],

 );

}

}

O resultado do código é esse: