

Flutter

From Rookie to Rock Star

ou

○ Que Aprendi em 20 dias

Quem sou eu?



Felipe Rocha
Mobile Software Engineer @ PagSeguro

Antes de tudo...

O que é o Dart?

Dart

Linguagem de programação criada pelo Google para o desenvolvimento de apps para Android, iOS, IoT, Web, ...

Características

Produtividade
Rápido
Portável
Acessível
Reativo

```
class Bicycle {  
  int cadence;  
  int _speed = 0;  
  int get speed => _speed;  
  int gear;  
  ...  
}
```

```
class Bicycle {  
  ...  
  Bicycle(this.cadence, this.gear);  
  
  void applyBreak(int decrement) {  
    _speed -= decrement;  
  }  
  
  void speedUp(int increment) {  
    _speed += increment;  
  }  
  ...  
}
```

```
void main() {  
  var bike = Bicycle(2, 1);  
  bike.speedUp(10);  
  bike.applyBreak(20);  
  print(bike);  
}
```

```
class Bicycle {  
  ...  
  @override  
  String toString() => 'Bicycle: $_speed km/h';  
}
```

Características

Produtividade

Rápido

Portátil

Acessível

Reativo

```
class Bicycle {  
    int cadence;  
    int _speed = 0;  
    int get speed => _speed;  
    int gear;  
    ...  
}
```

```
class Bicycle {  
    ...  
    Bicycle(this.cadence, this.gear);  
  
    void applyBreak(int decrement) {  
        _speed -= decrement;  
    }  
  
    void speedUp(int increment) {  
        _speed += increment;  
    }  
    ...  
}
```



```
class Bicycle {  
    ...  
    @override  
    String toString() => 'Bicycle: $_speed km/h';  
}
```

```
void main() {  
    var bike = Bicycle(2, 1);  
    bike.speedUp(10);  
    bike.applyBreak(20);  
    print(bike);  
}
```

Mas e agora...

O que é o Flutter?

Flutter

É um framework open source criado pelo Google para o desenvolvimento de app multiplataforma (Android e iOS)

Primeira release: Maio de 2017

Versão atual: 1.7.8+hotfix.3

Baseado no Dart

Características

React-style: estilo declarativo de escrever código

Hot-reload

UI podem ser construídas com Material Design e/ou Cupertino, APIs de Motion Design, scroll "natural"

Showcases



Google Ads



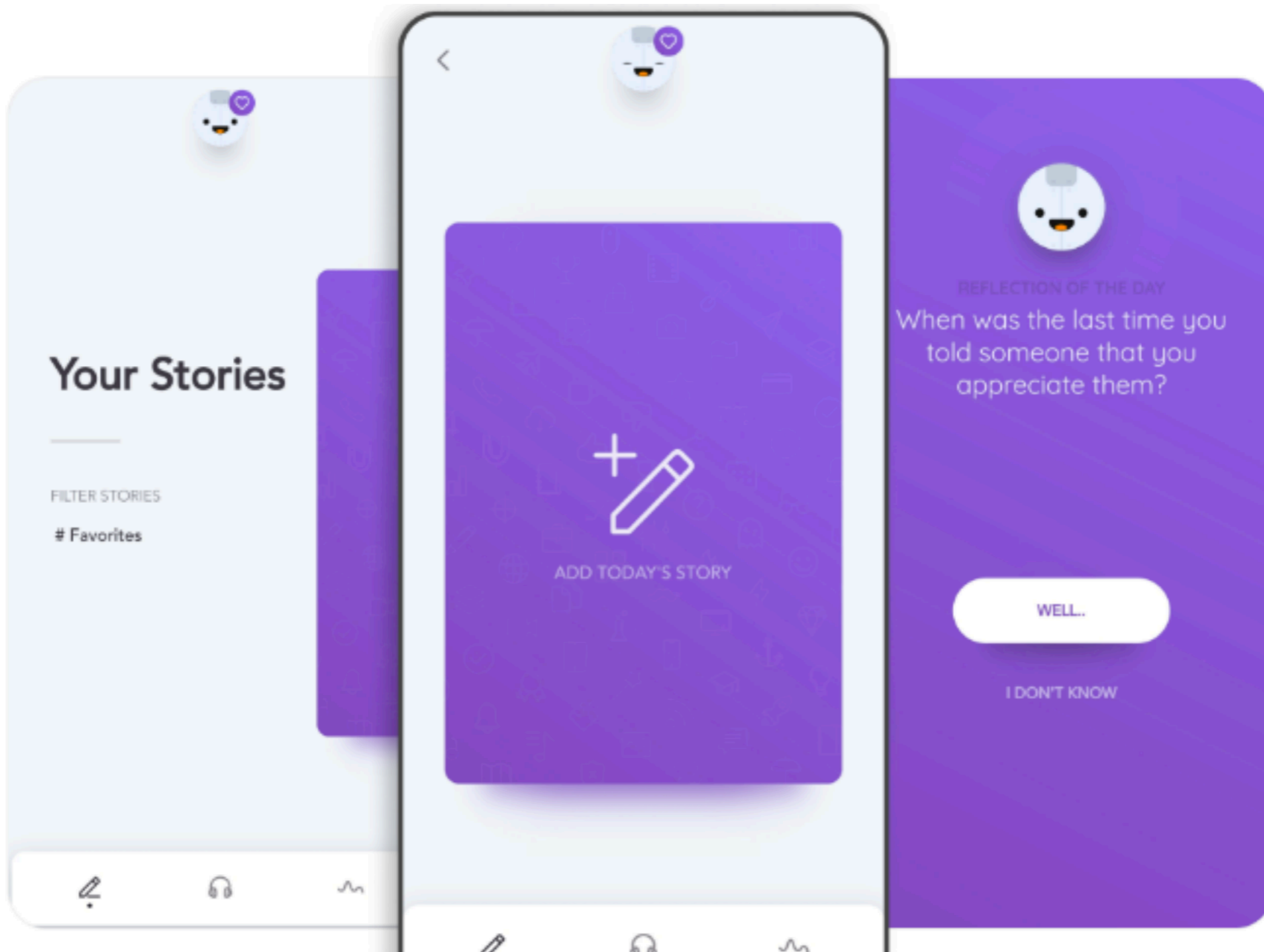
Alibaba

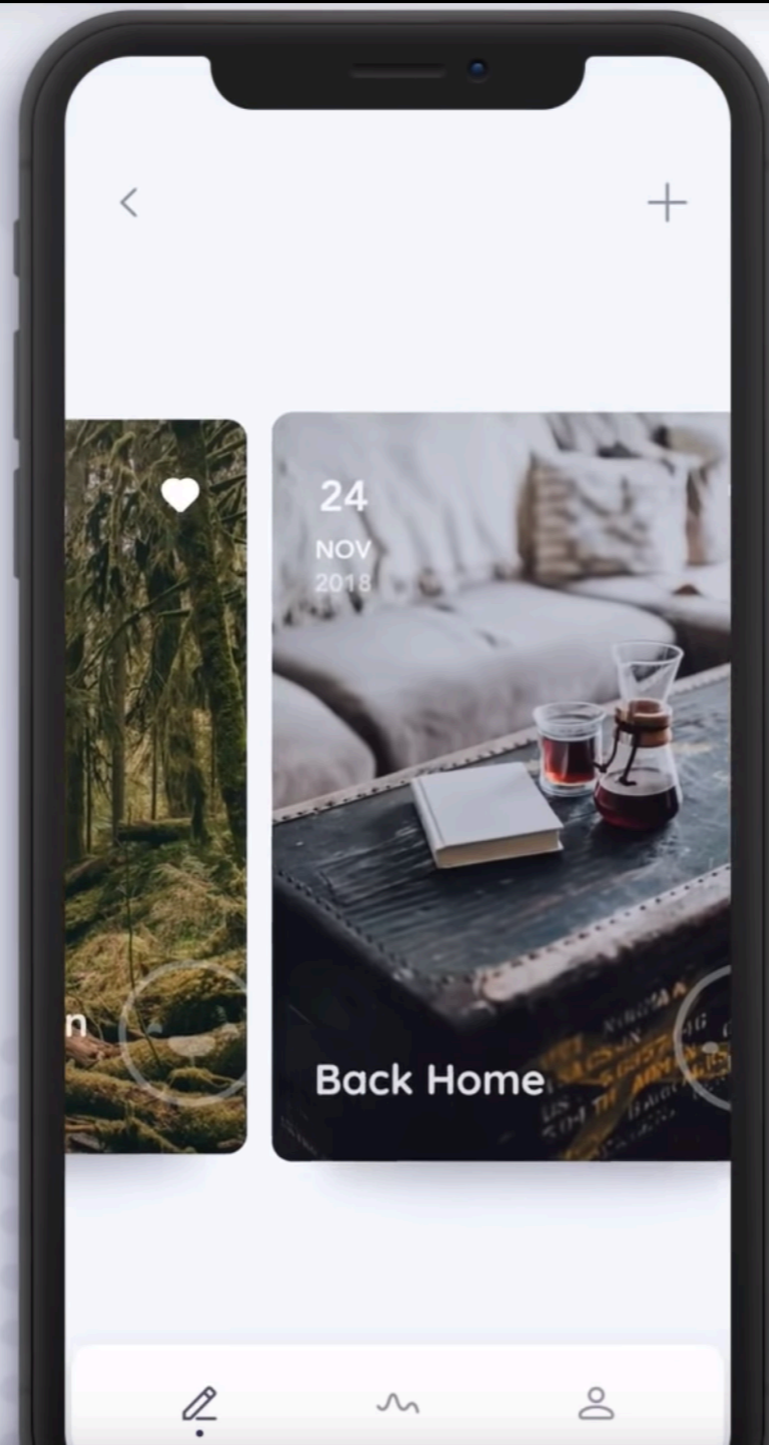


Reflectly

Tencent 腾讯

Tencent





<https://youtu.be/bB7jrSN0-d0>

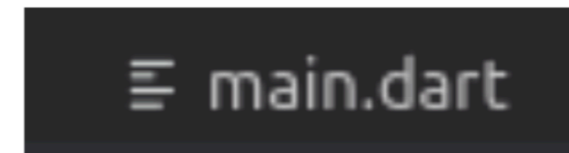
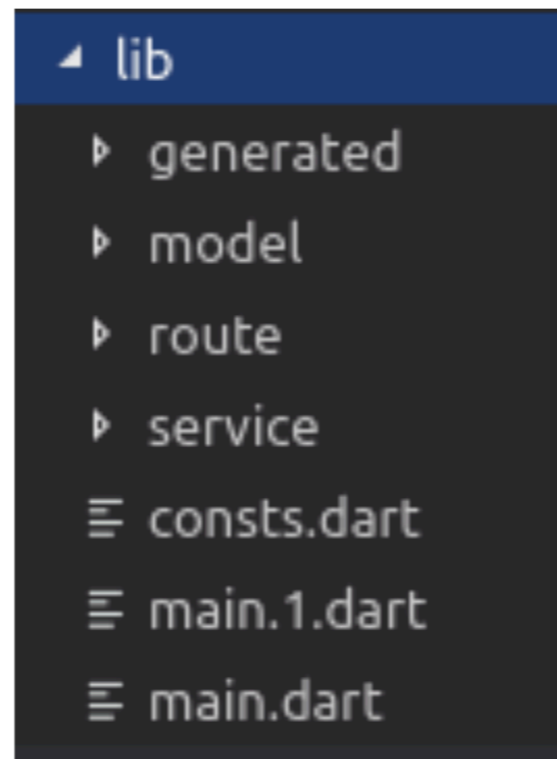
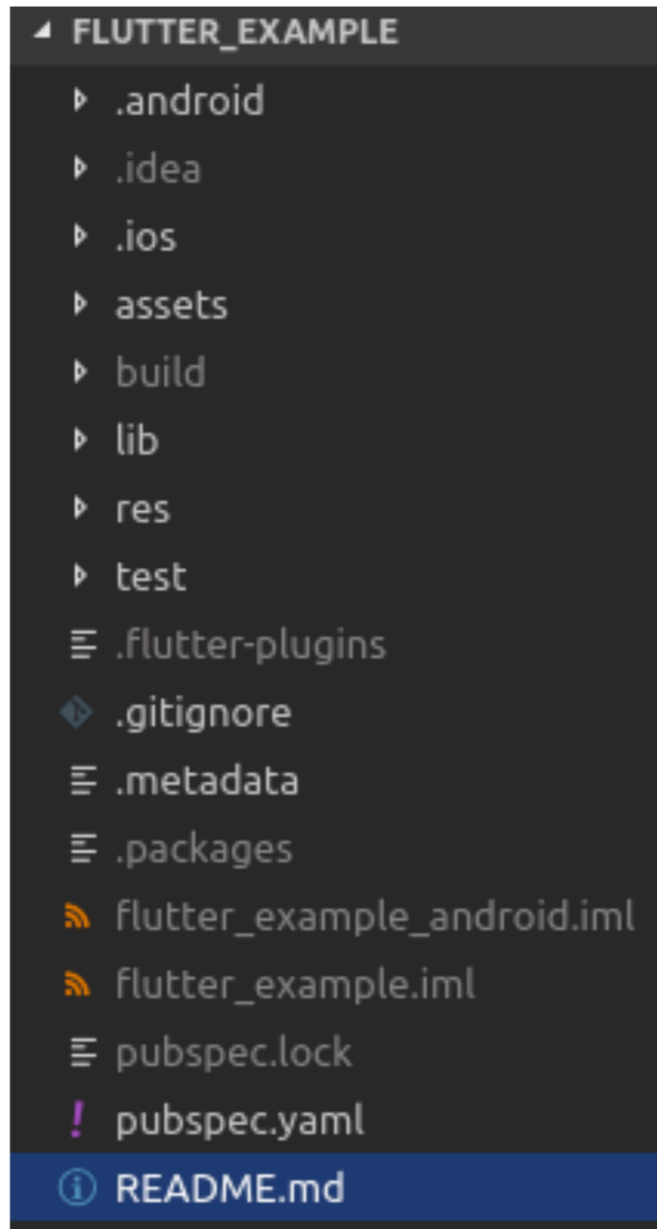
E agora, como começar?

flutter create *flutter_example*

cd *flutter_example*

flutter run






```
import 'package:flutter/material.dart'

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ), // ThemeData
      home: MyHomePage(title: 'Flutter Demo Home Page'),
    ); // MaterialApp
  }
}
```

Importante!

Todo app tem um começo!

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

Mais importante ainda!

Tudo é Widget!

Widgets

StatelessWidget

StatefulWidget

Text, TextField

Column, Row

Container

Center, Padding, Margin

ListView, GridView

Entre outros...

StatelessWidget vs. StatefulWidget

As minhas views serão estáticas? **StatelessWidget**

Preciso alterar o estado (valor, comportamento etc.) das minhas views em tempo de execução? **StatefulWidget**

State: entidade responsável por realizar as trocas de estados em um StatefulWidget

E como faço os layouts?

```
class ExampleWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Sample App"),
      ), // AppBar
      body: Center(
        child: MaterialButton(
          onPressed() {},
          child: Text('Hello'),
          padding: EdgeInsets.only(left: 10.0, right: 10.0),
        ), //MaterialButton
      ) // Center
    ); // Scaffold
  }
}
```

It's Showtime!

Proposta

Reescrever um dos fluxos
do app da firma

Fluxo escolhido

Recarga de

Celular Pré-Pago

Por qué?

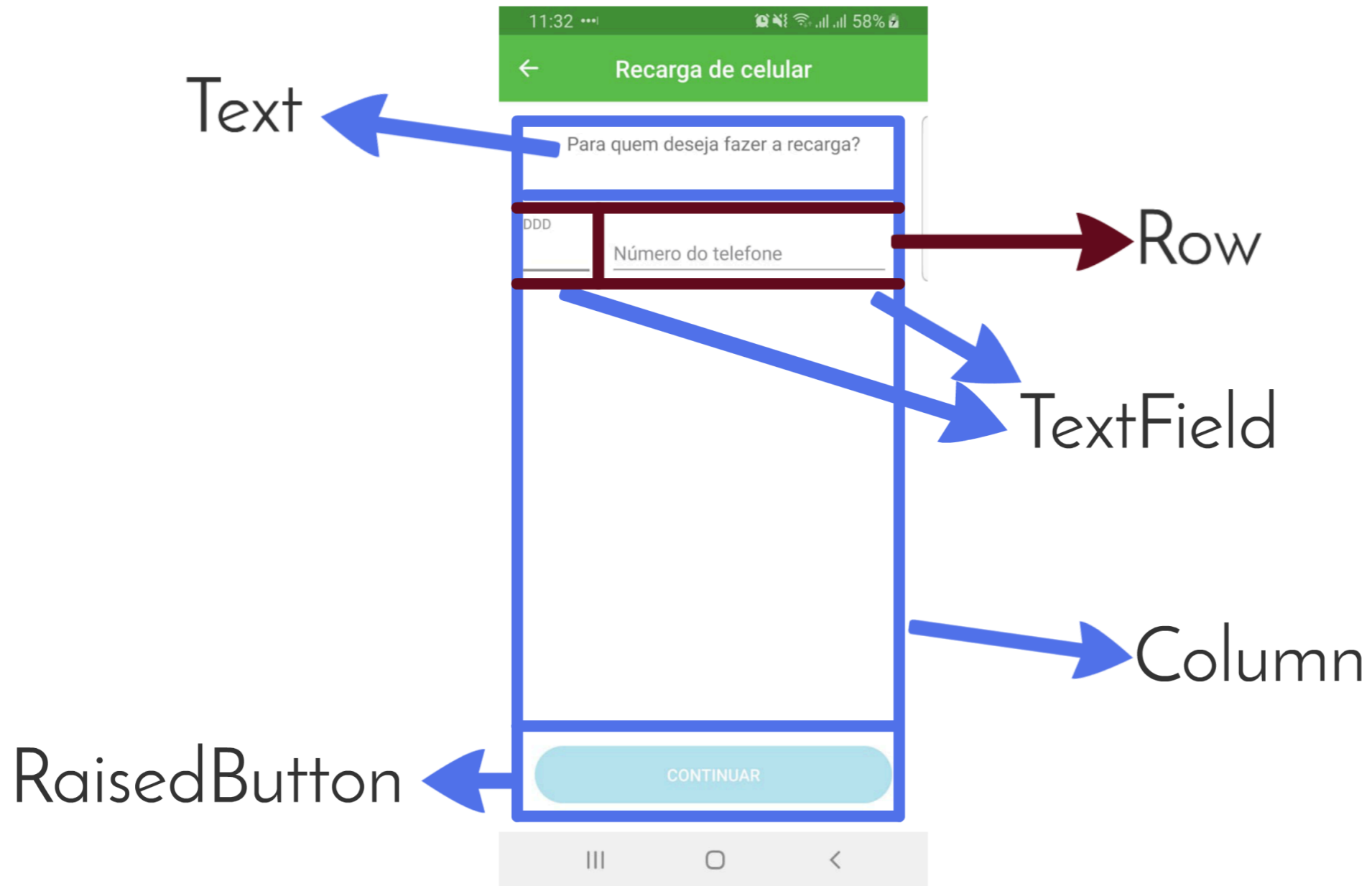
Não há integração
com sensores

Não há animações
complexas

Há várias telas/passos

Há requisições de rede

Integração com o código nativo



Comunicação de Rede e Assincronismo

Future/FutureBuilder

Future

Objeto que representa
uma computação "tardia"

```
Future<SummaryResponse> getSummary() async {  
    ...  
}
```


FutureBuilder

Widget que é construído a partir do resultado de um Future

```

class _MyHomePageState extends State<MyHomePage> {
  Future<SummaryResponse> _getSummaryResponse() {
    return getSummary();
  }

  @override
  Widget build(BuildContext context) {
    return FutureBuilder<SummaryResponse>(
      future: _getSummaryResponse(),
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.done) {
          if (snapshot.hasData) {
            return NumberRechargePhoneWidget();
          } else {
            return Center(
              child: Text("${snapshot.error}"),
            );
          }
        } else {
          return Center(
            child: Material(
              child: CircularProgressIndicator(),
            ),
          );
        }
      });
  }
}

```

async/await

async

Indica que uma
função/método é
assíncrono

await

A operação que a procede
será executada e o código que o
chamou esperará o seu resultado
para continuar

```
Future<SummaryResponse> getSummary() async {  
  ...  
  var client = _getHttpClient();  
  
  final response = await client.get('$urlSummary', headers: {  
    'X-Token': '$token',  
  }).whenComplete(() => client.close());  
  
  ...  
  return summaryResponse;  
}
```

Comunicação entre o Flutter e o Hospedeiro

Method Channel

```
public class MainActivity extends FlutterActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        GeneratedPluginRegistrant.registerWith(this);  
        ...  
    }  
}
```

```
const platform = const MethodChannel('flutter/recharge_phone');
```

```
Future<String> getFormattedUrl(String url) async {  
    String formattedUrl = '';  
    String method = 'getFormattedUrl';  
    try {  
        formattedUrl = await platform.invokeMethod(method, data);  
    } on PlatformException catch (e) {  
        debugPrint(e.message);  
    }  
    return formattedUrl;  
}
```

```
Future<SummaryResponse> getSummary() async {  
    String urlSummary = '';  
    String token = '';  
  
    try {  
        token = await platform.invokeMethod('getApplicationToken');  
        urlSummary = await platform.invokeMethod('getRechargePhoneSummary');  
    } on PlatformException catch (e) {  
        debugPrint(e.message);  
    }  
    ...  
}
```

```
new MethodChannel(getFlutterView(), CHANNEL).setMethodCallHandler(  
    new MethodCallHandler() {  
        @Override  
        public void onMethodCall(MethodCall call, Result result) {  
            switch (call.method) {  
                ...  
                case FORMATTED_URL:  
                    result.success((String) call.arguments);  
                    break;  
                ...  
                case SUMMARY:  
                    result.success(BASE_URL + "/summary");  
                    break;  
                ...  
                default:  
                    result.notImplemented();  
            }  
        }  
    });
```

```
public class MainActivity extends FlutterActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        GeneratedPluginRegistrant.registerWith(this);  
        ...  
    }  
}
```



```
const platform = const MethodChannel('flutter/recharge_phone');
```

```
Future<String> getFormattedUrl(String url) async {  
    String formattedUrl = '';  
    String method = 'getFormattedUrl';  
    try {  
        formattedUrl = await platform.invokeMethod(method, data);  
    } on PlatformException catch (e) {  
        debugPrint(e.message);  
    }  
    return formattedUrl;  
}
```

```
Future<SummaryResponse> getSummary() async {  
    String urlSummary = '';  
    String token = '';  
  
    try {  
        token = await platform.invokeMethod('getApplicationToken');  
        urlSummary = await platform.invokeMethod('getRechargePhoneSummary');  
    } on PlatformException catch (e) {  
        debugPrint(e.message);  
    }  
    ...  
}
```

```
new MethodChannel(getFlutterView(), CHANNEL).setMethodCallHandler(
    new MethodCallHandler() {
        @Override
        public void onMethodCall(MethodCall call, Result result) {
            switch (call.method) {
                ...
                case FORMATTED_URL:
                    result.success((String) call.arguments);
                    break;
                ...
                case SUMMARY:
                    result.success(BASE_URL + "/summary");
                    break;
                ...
                default:
                    result.notImplemented();
            }
        }
    }
}
```

Queremos mudar de tela...
E agora?

Navegar é preciso!

Navigation

Controla a navegação entre telas, com **push** e **pop** como operações principais

Push pode ser feito
pela classe ou pelo
identificador


```
Navigator.pushNamed(context, CarrierRoute.tag);
```

```
void main() {  
    // debugPaintSizeEnabled = true;  
    runApp(new MaterialApp(  
        theme: getDefaultTheme(),  
        title: '',  
        home: InputNumberRoute(),  
        routes: {  
            InputNumberRoute.tag: (context) => InputNumberRoute(),  
            CarrierRoute.tag: (context) => CarrierRoute(),  
            ValuesRoute.tag: (context) => ValuesRoute(),  
            ResumeRoute.tag: (context) => ResumeRoute(),  
            ReceiptRoute.tag: (context) => ReceiptRoute(),  
        }));  
}
```

E como eu importo as libs?
E as minhas imagens?

pubspec.yaml

Dependências de desenvolvimento e produção

Declaração dos recursos

Versionamento do aplicativo

Declaração do bundle


```
name: flutter_example
description: A new flutter module project.
...
dependencies:
  flutter:
    sdk: flutter

 /cupertino_icons: ^0.1.2
  http: ^0.11.3+17
  flutter_masked_text: ^0.8.0
  http_middleware: ^1.0.0
  http_logger: ^1.0.0
  cached_network_image: ^1.0.0
  intl: ^0.15.8
  after_layout: ^1.0.7+2

dev_dependencies:
  flutter_test:
    sdk: flutter
```

...

flutter:

uses-material-design: true

assets:

- assets/images/ic_success.png

module:

androidPackage: br.com.uol.ps.myaccount.phone_recharge

iosBundleIdentifier: br.com.uol.ps.myaccount.phoneRecharge

...

E como integro com o código já existente?

Exemplo feito com Android,
mas iOS é bem parecido!

```
// MyApp/settings.gradle
include ':app' // assumed existing content
setBinding(new Binding([gradle: this])) // new
evaluate(new File( // new
    settingsDir.parentFile, // new
    'recharge_phone/.android/include_flutter.groovy' // new
)) // new
```

```
// MyApp/app/build.gradle
..
dependencies {
    implementation project(':flutter')
    ..
}
```

E no final das contas,
como ficou?

Recarga de celular

Para quem deseja fazer a recarga?

DDD
11

Número do telefone
9876

CONTINUAR

1 2 3 ✕

4 5 6 OK

7 8 9 .-

0 @

https://youtu.be/p-FkKBO0_-E

E os próximos 20 dias?

Acessibilidade

Animações

Arquitetura

Bloc

MVP

MVVM

Temas entre Android e iOS

Testes

Links Úteis

DartPad - <https://dartpad.dartlang.org>
Playground de Dart

JSON 2 Dart - https://javiercbk.github.io/json_to_dart/
Criação de classe Dart a partir de um JSON

Flutter for Android developers - <https://flutter.dev/docs/get-started/flutter-for/android-devs>
Guia básico para quem vem do Android

Perfil do Flutter no Twitter - <https://twitter.com/flutterdev>

Perfil Flutter Weekly - <https://twitter.com/FlutterWk>
Newsletter semanal

Material Design para Flutter - <https://material.io/develop/flutter>

Flutter oficial - <https://flutter.dev>

Dart oficial - <https://dartlang.org>

Documentação oficial da API - <https://docs.flutter.io>

Codelabs - <https://flutter.dev/docs/codelabs>

Roadmap de Flutter - <https://github.com/flutter/flutter/wiki/Roadmap>

Obrigado!