

[< VOLTAR](#)

Integração com código nativo

Muitas vezes precisamos acessar algum recurso específico de hardware do celular que ou algum outro App que é desenvolvido de forma nativa em nossa Aplicação Flutter. As vezes há recursos específicos da Apple (iOS) ou Android cujas chamadas são feitas de formas distintas. Para isso, precisamos utilizar um padrão de comunicação entre nosso App desenvolvido em Flutter e o App nativo e essa aula mostra como fazer isso.

NESTE TÓPICO

- › Plataform Channel
- › Criação da interface do usuário ? Flutter
- › Projeto nativo
- › Usando o plataform cha no Flutter
- › Download do projeto
- › Referências



Plataform Channel

O **Plataform Channel** (canal de plataforma) é um mecanismo de comunicação entre a aplicação híbrida e a aplicação nativa.

É muito comum precisamos desenvolver algo que seja muito específico para Android e iOS, especialmente quando usamos bibliotecas de terceiros que não possuem suporte ao Flutter, apenas pra, por exemplo, o Java (nativo Android), o Kotlin (nativo android) ou o Swift (nativo iOS). Isso ocorre pois o Flutter de fato ainda é muito novo.

Para contornar qualquer impossibilidade de comunicação com o Android, os desenvolvedores do Flutter criaram o **Plataform Channel** que permite chamadas de métodos através de um canal de comunicação entre as aplicações nativas e híbridas.

O vídeo abaixo mostra a ideia dessa comunicação baseada na documentação oficial do Flutter. Não deixe de assistir cuidadosamente e ler, também, a documentação oficial no site do Flutter.



O que é o "Method Channel"



VEJA A DOCUMENTAÇÃO COMPLETA SOBRE ESTE ASSUNTO

Clique [aqui](https://flutter.dev/docs/development/platform-integration/platform-channels) (<https://flutter.dev/docs/development/platform-integration/platform-channels>) para ler a documentação oficial do Flutter sobre este assunto.



Criação da interface do usuário ? Flutter

Vamos começar a entender este processo criando uma pequena e bastante simples aplicação que mostra essa integração usando o **platform channel**.

O vídeo abaixo apresenta o aplicativo que iremos desenvolver nessa aula e, ao mesmo tempo, mostra a criação da interface de usuário principal do lado do desenvolvimento híbrido, ou seja, em Flutter.

Veja cuidadosamente o vídeo pois todo detalhe é importante neste momento.

Interface híbrida (Flutter)



Projeto nativo

Agora que já desenvolvemos a interface de usuário em Flutter, podemos começar a desenvolver o lado nativo, neste caso feito em Java.

Lembre-se que o foco dessa disciplina não é apresentar o Java, por se tratar de um desenvolvimento nativo apenas para Android e, por isso desenvolvemos este aplicativo utilizando um pacote externo, conforme explicado no vídeo abaixo.



Desenvolvimento nativo - Java



Durante o desenvolvimento deste vídeo, um bloco de código precisou ser injetado para correção do arquivo de manifesto do Android. Abaixo você encontra o bloco completo que foi necessário ser incluído para essa integração.

```
1. <activity
2.     android:name=".MainActivity"
3.     android:launchMode="singleTop"
4.     android:theme="@style/LaunchTheme"
5.     android:configChanges="orientation|keyboardHidden|keyboard|screenSize|sm
alllestScreenSize|locale|layoutDirection|fontScale|screenLayout|density|uiMode"
6.     android:hardwareAccelerated="true"
7.     android:windowSoftInputMode="adjustResize">
8.     <meta-data
9.         android:name="io.flutter.embedding.android.NormalTheme"
10.        android:resource="@style/NormalTheme"
11.    />
12.    <meta-data
13.        android:name="io.flutter.embedding.android.SplashScreenDrawable"
14.        android:resource="@drawable/launch_background"
15.    />
16.    <intent-filter>
17.        <action android:name="android.intent.action.MAIN"/>
18.        <category android:name="android.intent.category.LAUNCHER"/>
19.    </intent-filter>
20. </activity>
```

Usando o plataform channel no Flutter

Agora que nossa aplicação já está prontinha do lado nativo, vamos começar a desenvolver a integração no Flutter.

O vídeo abaixo mostra como fazer isso cuidadosamente. Assiste pausadamente o vídeo para entender cada uma das etapas descritas no processo de desenvolvimento da aplicação.



Integração do Plataform Channel no Flutter



E agora, para finalizarmos nosso aplicativo, podemos criar o método que mostra a caixa de diálogo quando o usuário clica no botão para zerar o contador, conforme o vídeo abaixo.

Zerando o contador



Download do projeto

Caso você precise baixar o projeto completo para qualquer tipo de consulta ou análise, basta clicar no arquivo abaixo para download do mesmo, compactado. O projeto já inclui a correção do arquivo de manifesto do Android nativo e a imagem de recurso utilizada durante o desenvolvimento.



[Download do projeto completo](#)

Quiz

Exercício Final

Integração com código nativo

INICIAR ➤

Referências

DART. **Dart documentation**. *Site*. Disponível em: <https://dart.dev/>. Acesso em: 08 dez. 2020.

MATERIAL DESIGN. **Material Design documentation**. *Site*. Disponível em: <https://material.io/>. Acesso em: 08 dez. 2020.

FLUTTER. **Flutter docs**. *Site*. Disponível em: <https://flutter.dev>. Acesso em: 09 dez.2020.

WINDMILL, Eric. **Flutter in action**. Nova Iorque: Manning publications, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/flutter-in-action/9781617296147/>. Acesso em: 08 dez. 2020.

SINHA, Sanjib. **Quick start guide to Dart programming**: create high performance applications for the web and mobile. Lompoc, CA, EUA: Apress, 2019. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/quick-start-guide/9781484255629/>. Acesso em: 09 dez. 2020.

ALESSANDRIA, Simone. **Flutter projects**: a practical, project-based guide to building real-words cross-platform mobile applications and games. Birmingham, Reino Unido: Packt Publishing, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/flutter-projects/9781838647773/>. Acesso em: 09 dez. 2020.

ZACCAGNINO, Carmine. **Programming Flutter**. [s.l.]: The Pragmatic Bookshelf, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/programming-flutter/9781680507621/>. Acesso em: 09 dez. 2020.



Avalie este tópico



ANTERIOR

<

Índice

≡

Página?

>

Gerenciamento de estados com Redux

(https://ava.uninove.br/curso=)

© Todos os direitos reservados

Anteções complexas com AnimatedBuilder

Biblioteca

rAnteções complexas com AnimatedBuilder

(https://www.uninove.br/conheca-
a-
uninove/biblioteca/sobre-
a-
biblioteca/apresentacao/)
Portal Uninove
(http://www.uninove.br)
Mapa do Site

