

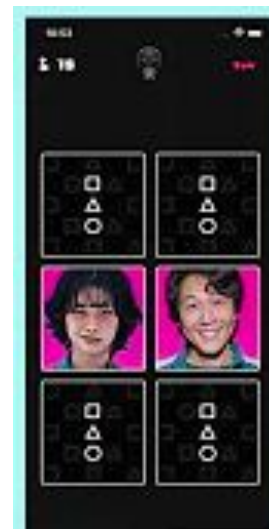


Flutter Avançado - Aula 5

Exercícios 13 a 15



- Jogo da Memória em Flutter [Ver mais](#)
- Plugin: Pigeon [Ver mais](#)





- Como o Flutter é uma tecnologia multiplataforma, em alguns momentos precisaremos trabalhar com a plataforma nativa
- De acordo com cada plataforma, precisaremos acessar o código na linguagem nativa do dispositivo, seja ele *mobile* ou *desktop*
- Platform Channels disponibilizam mecanismos para que possamos nos comunicar entre a plataforma nativa e o nosso código dart





Plataforma	Linguagens
iOS	Swift ou Objective-C
Android	Kotlin ou Java
macOS	Swift ou Objective-C
Windows	C++
Linux	C



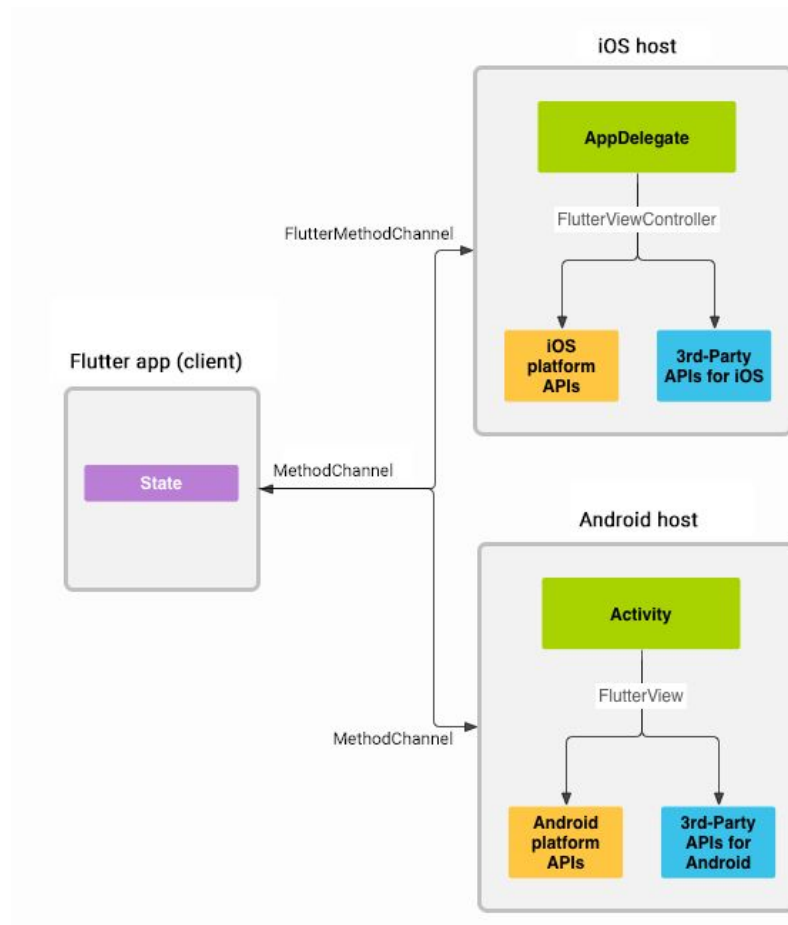
Determinando a plataforma atual

```
String _getPlataformName() {  
  if (kIsWeb)  
    return "Web";  
  
  if (Platform.isAndroid)  
    return "Android";  
  
  if (Platform.isFuchsia)  
    return "Fuchsia";  
  
  if (Platform.isIOS)  
    return "iOS";  
  
  if (Platform.isLinux)  
    return "Linux";  
  
  if (Platform.isMacOS)  
    return "MacOS";  
  
  if (Platform.isWindows)  
    return "Windows";  
  
  return "";  
}
```

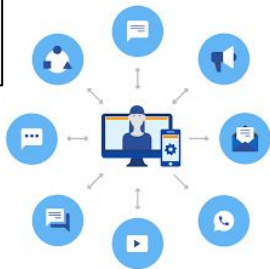




- O MethodChannel possibilita o envio e recebimento de mensagens entre o código nativo e o Flutter



```
class MethodChanelTest {  
    final String _channelID = "proway/methodChannel";  
    MethodChannel? _plataform = null;  
  
    MethodChanelTest() {  
        this._plataform = MethodChannel(this._channelID);  
    }  
  
    Future<String> getMessage() async {  
        return await this._plataform!.invokeMethod("getMessage");  
    }  
  
    Future<String> getMessageWithParams(String prUserName) async {  
        return await this._plataform!.invokeMethod("getMessageWithParams", {  
            "name": prUserName  
        });  
    }  
}
```



```
package com.example.plataform_channel

import androidx.annotation.NonNull
import io.flutter.embedding.android.FlutterActivity
import io.flutter.embedding.engine.FlutterEngine
import io.flutter.plugin.common.MethodChannel

class MainActivity: FlutterActivity() {
    override fun configureFlutterEngine(@NonNull flutterEngine: FlutterEngine) {
        super.configureFlutterEngine(flutterEngine)

        MethodChannel(
            flutterEngine.dartExecutor.binaryMessenger,
            "proway/methodChannel"
        ).setMethodCallHandler { call, result ->

            if (call.method == "getMessage")
                result.success("Teste 123");

            if (call.method == "getMessageWithParams") {
                val message: String? = call.argument("name")

                result.success("Teste 123 - " + message);
            }
        }
    }
}
```





- EventChannels possibilitam o envio e recebimento de streams entre o código nativo e o código dart
- Isso é útil quando precisamos receber informações de sensores, como giroscópios e acelerômetros
- Usado para receber informações a partir de eventos no dispositivo nativo



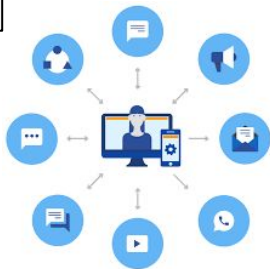
```
import 'package:flutter/services.dart';

class EventChannelTest {
  final String _channelID = "proway/eventChannel";
  EventChannel? _plataform = null;

  EventChannelTest() {
    this._plataform = EventChannel(this._channelID);
  }

  EventChannel get plataform => this._plataform!;
}
```

```
child: StreamBuilder(
  stream: this._channelTest.plataform.receiveBroadcastStream(),
```



```
EventChannel(flutterEngine.dartExecutor.binaryMessenger,
"proway/eventChannel").setStreamHandler(
    object : EventChannel.StreamHandler {
        override fun onListen(args: Any?, events: EventChannel.EventSink) {
            var handler = Handler(Looper.getMainLooper())
            handler.postDelayed(object : Runnable {
                override fun run() {
                    val sdf = SimpleDateFormat("dd/M/yyyy hh:mm:ss")
                    val currentDate = sdf.format(Date())
                    events.success(currentDate)
                    handler.postDelayed(this, 1000)
                }
            }, 0)
        }

        override fun onCancel(arguments: Any?) {
            println("cancelling listener")
        }
    }
)
```



- Não é possível utilizar platform channel em aplicações web
- Porém, é possível realizar chamadas javascript diretamente no código dart
- Ambas as tecnologias são compatíveis, portanto, é possível ter acesso a funções e outros recursos oriundos do navegador



Criando um arquivo .js com as funções que serão chamadas

```
function jsNotifier(message) {  
    console.log(message);  
  
    alert(message);  
}
```

Adicionando o fonte criado ao meu arquivo index.html

```
<script src="index.js" defer></script>
```





Realizando a chamada do código javascript, no Dart

```
import 'dart:js' as js;  
  
...  
  
js.context.callMethod("jsNotifier", ["Teste de Mensagem"]);
```





- Vamos criar uma tela de autenticação, para não precisar mais fixar os dados do usuário
- Na tela de login, o usuário deverá informar o nome do usuário
- A rotina deve validar se o nome do usuário foi preenchido
- E deve validar também se o usuário informado existe
- Se existir, deve ir para a página que lista os usuários e as conversas



Informe o nome do usuário

Nome de usuário



O campo nome do usuário é obrigatório



- Na tela de login, adicionaremos também uma opção para cadastrar novos usuários
- Na tela de cadastro, o usuário deverá informar o nome do usuário e o apelido do usuário
- A imagem pode ficar em branco ou preencher com uma imagem padrão
- Os campos de nome do usuário e apelido são obrigatórios
- Após cadastrar, o usuário poderá se logar ou já entrar na página inicial de chat



Informe o nome do usuário

Nome de usuário

Ainda não possui cadastro? Cadastre-se agora!

[Cadastrar-se](#)



← Novo usuário

Informe os dados de cadastro do usuário

Nome do usuário

Apelido do usuário





- Nosso próximo app será um app de administração de uma loja virtual
- Nesse primeiro exercício, vamos criar um novo projeto no firebase
- Crie também uma base de dados no Firestore e uma coleção de Produtos
- Nos documentos de produtos, teremos os campos de nome e preço do produto
- Além de criar o projeto no Firebase, crie um projeto configurado para poder acessar o projeto
- Crie uma página inicial para esse projeto, com um drawer

