



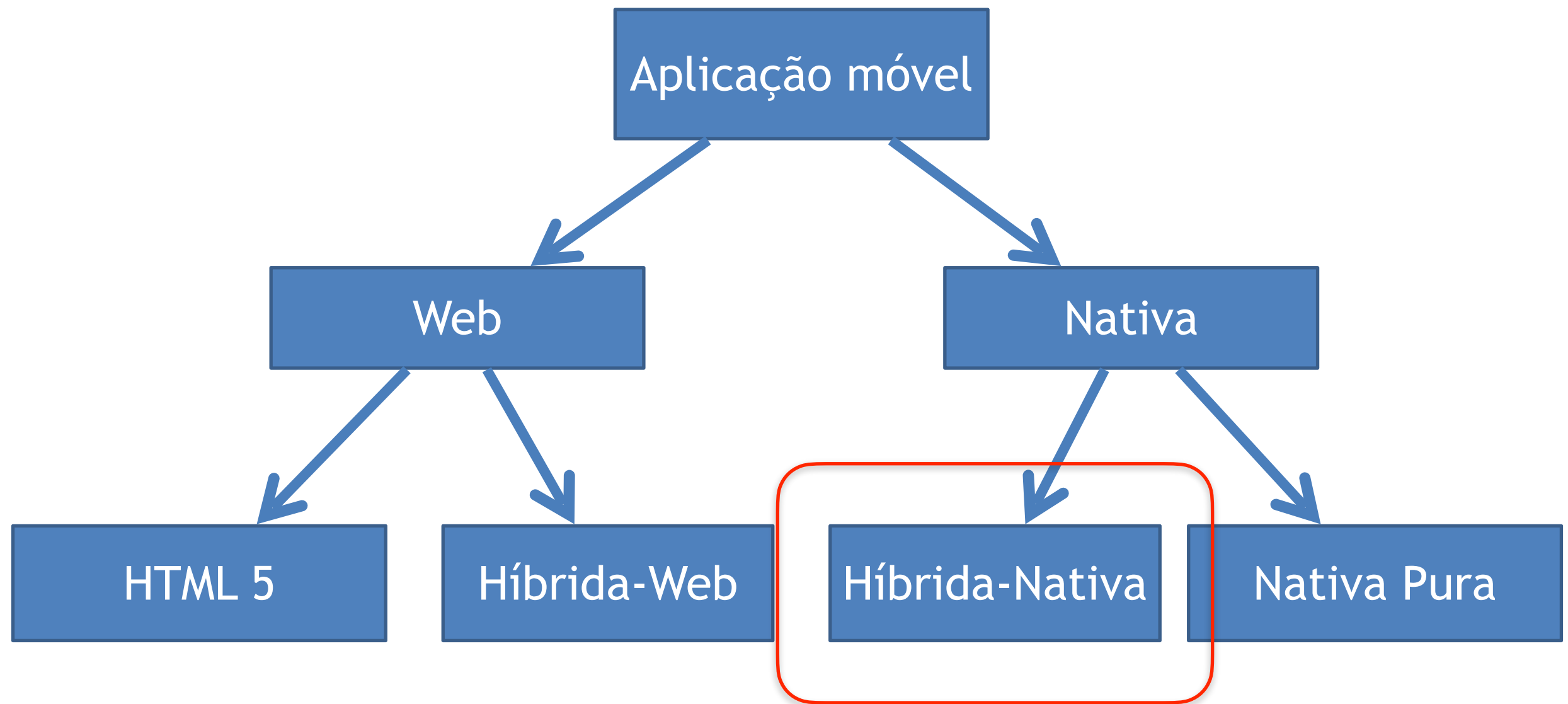
Híbrido - Nativo

Informações

Na próxima semana não haverá aula teórica (Carnaval), quer para os alunos de dia quer para os alunos do pós-laboral

- Será publicado um vídeo sobre arquitetura de aplicações Android
- Haverá quiz sobre a matéria do vídeo

Arquitetura de aplicações móveis



Híbrida-Nativa

Tal como as aplicações híbridas-web, são aplicações nativas instaláveis através da App store / Play store.

Tal como as aplicações híbridas-web, são desenvolvidas numa única linguagem.

Híbrida-Nativa

Tal como as aplicações híbridas-web, são aplicações nativas instaláveis através da App store / Play store.

Tal como as aplicações híbridas-web, são desenvolvidas numa única linguagem.

No entanto, não usam Webviews. Desenham os elementos gráficos de forma nativa, conseguindo uma aparência e fluidez similar às aplicações nativas puras

Híbrida-Nativa

Tal como as aplicações híbridas-web, são aplicações nativas instaláveis através da App store / Play store.

Tal como as aplicações híbridas-web, são desenvolvidas numa única linguagem.

No entanto, não usam Webviews. Desenham os elementos gráficos de forma nativa, conseguindo uma aparência e fluidez similar às aplicações nativas puras

Exemplos: Appcelerator Titanium, React Native, Flutter, .NET MAUI (antigo Xamarin), Unity (jogos), Kotlin Multiplatform

Comparação

Híbrida-Web



HTML

```
<button value="click me"/>
```

(interpretado pelo browser para
mostrar o botão,
o botão é virtual)

Comparação

Híbrida-Web

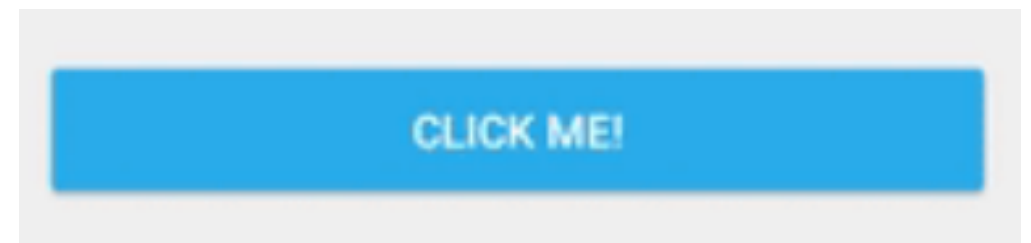


HTML

```
<button value="click me"/>
```

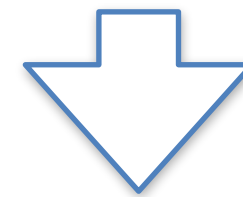
(interpretado pelo browser para
mostrar o botão,
o botão é virtual)

Híbrida-Nativa



React-Native

```
<Button title="click me" />
```



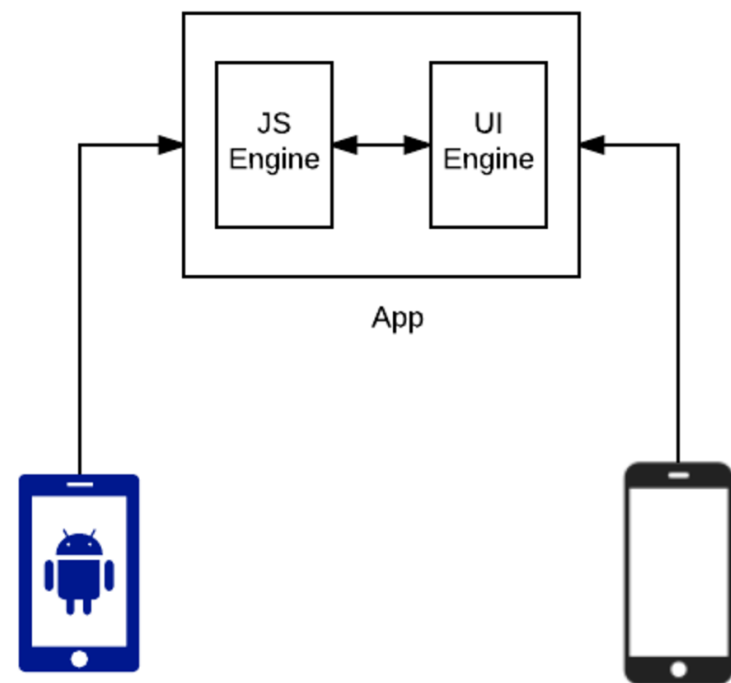
Android/Kotlin

```
Button("click me")
```

(cria um botão real através da
sistema operativo)

Híbrida-Nativa

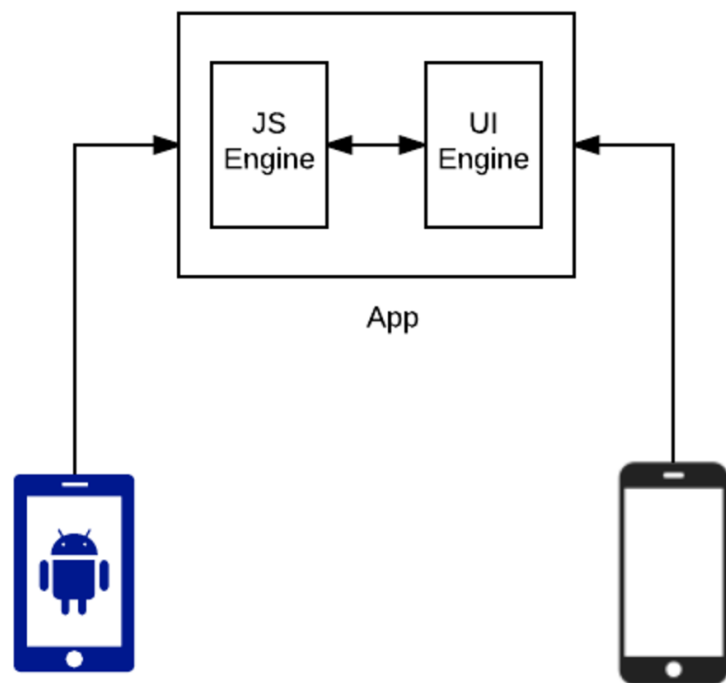
As aplicações híbridas-nativas usam uma destas duas técnicas:



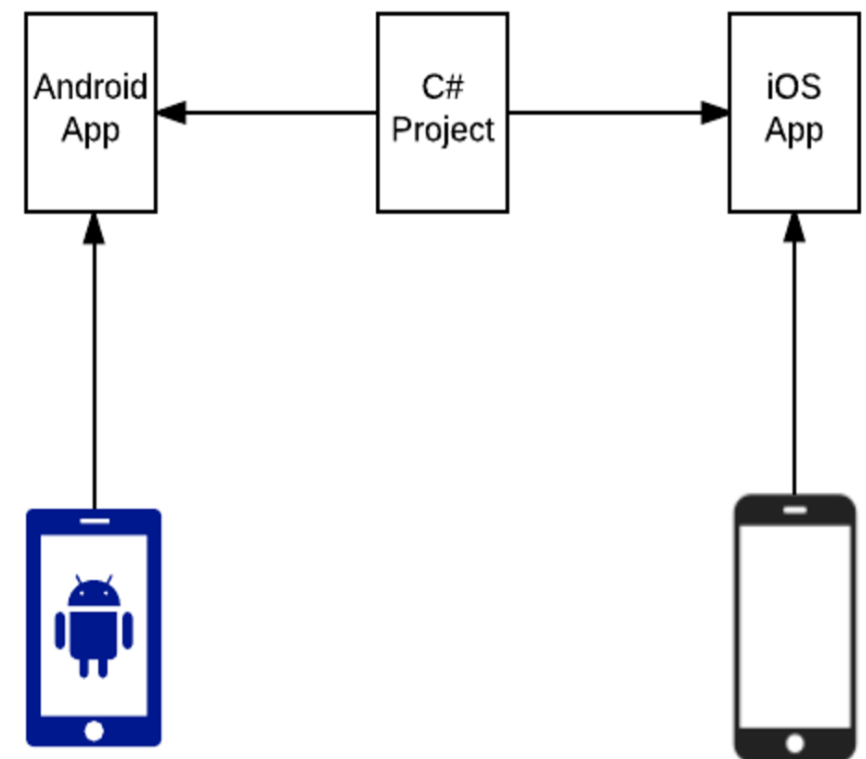
JS Engine - Faz a ponte entre componentes gráficos Javascript e os componentes gráficos nativos

Híbrida-Nativa

As aplicações híbridas-nativas usam uma destas duas técnicas:



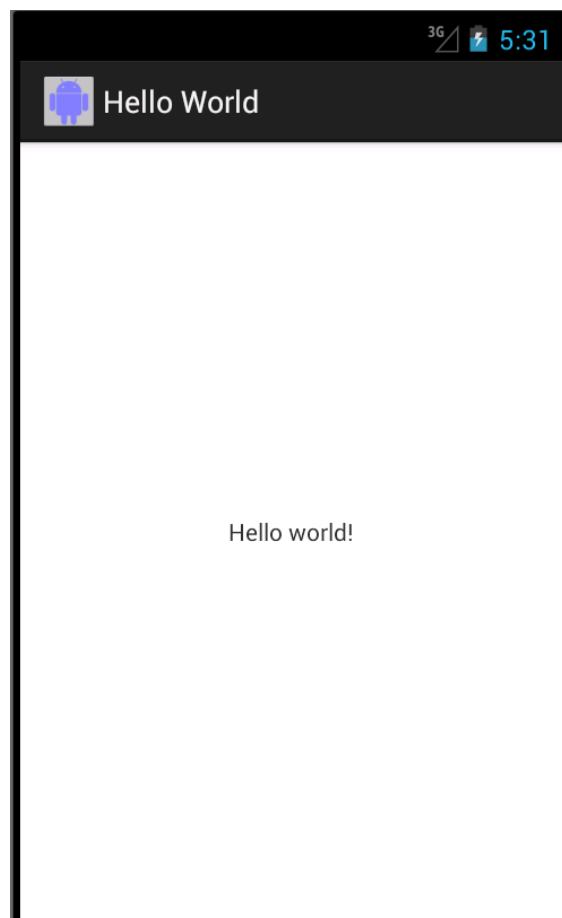
JS Engine - Faz a ponte entre componentes gráficos Javascript e os componentes gráficos nativos



Cross-compiler - Compila uma linguagem comum (Dart, C#) para as várias plataformas

Híbrida-Nativa

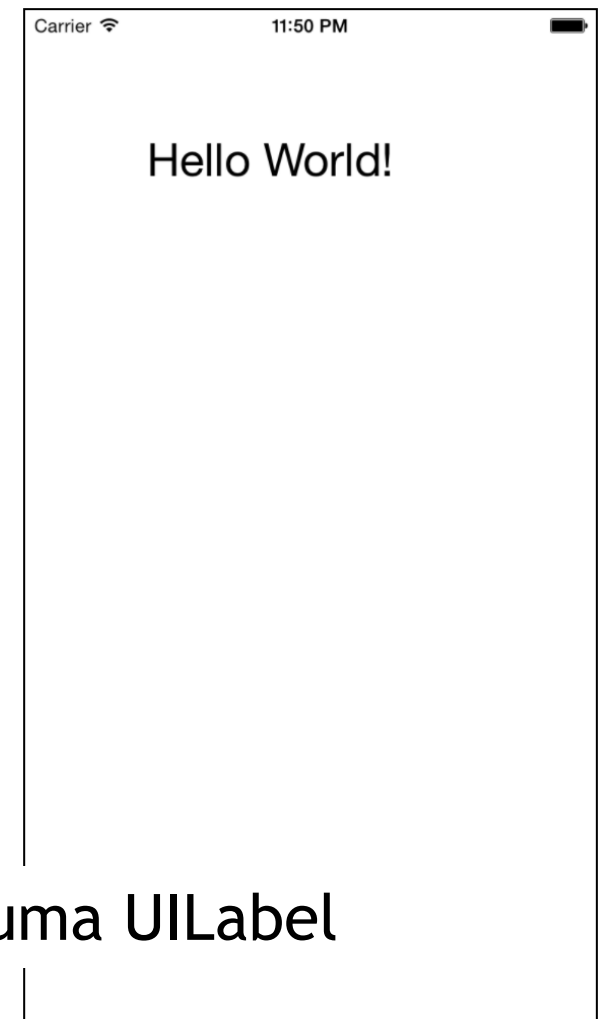
JS Engine (exemplo em React Native)



Android - cria uma TextView

```
<ReactNative.Text style={styles.text}>  
  Hello World  
</ReactNative.Text>
```

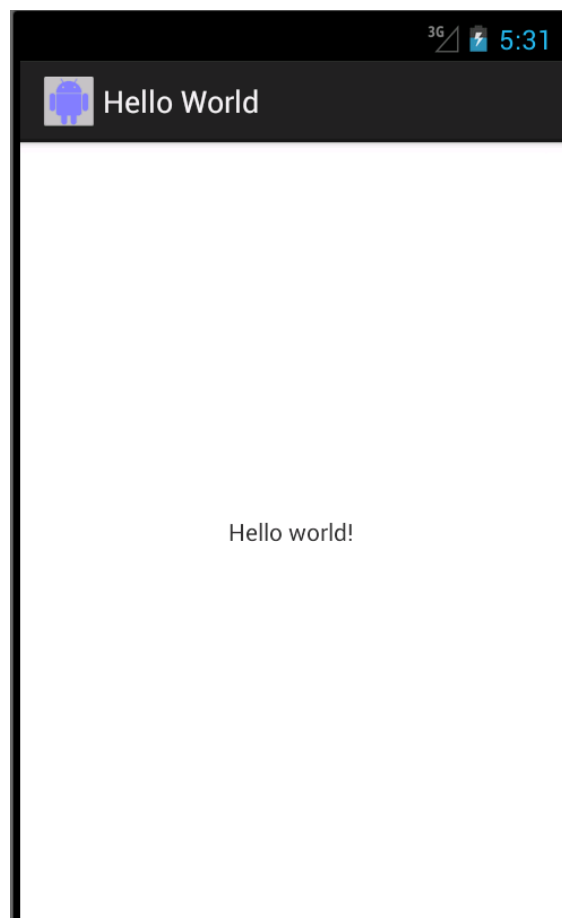
Esta transformação é feita em
tempo de execução
(runtime)



iOS - cria uma UILabel

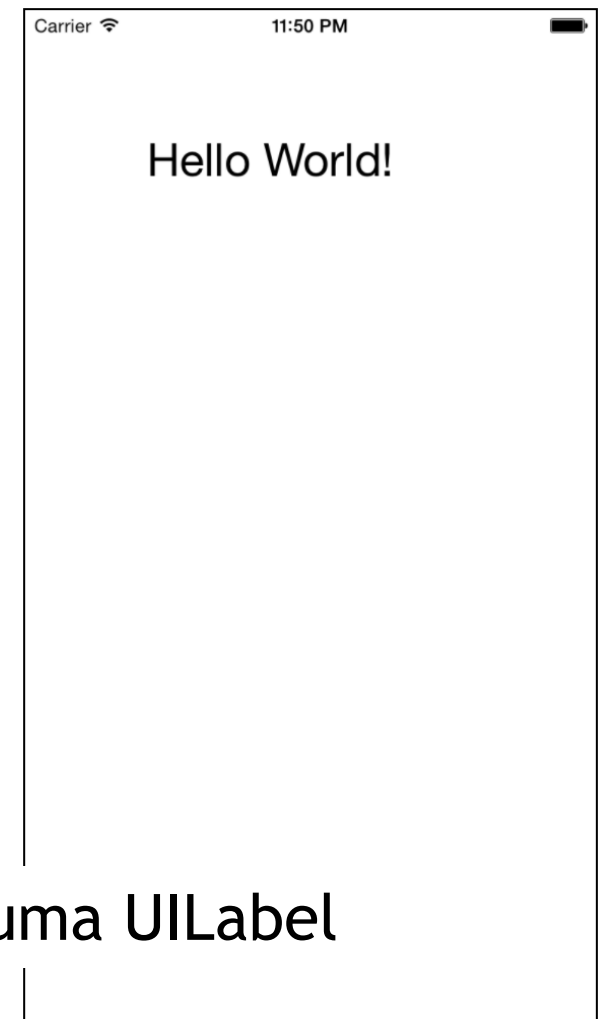
Híbrida-Nativa

Cross compiler (exemplo em .NET MAUI)



```
<ContentPage>
  <Label Text="Hello, World!"
        FontSize="32"
        HorizontalOptions="Center"/>
</ContentPage>
```

Esta transformação é feita em
tempo de compilação
(compile-time)



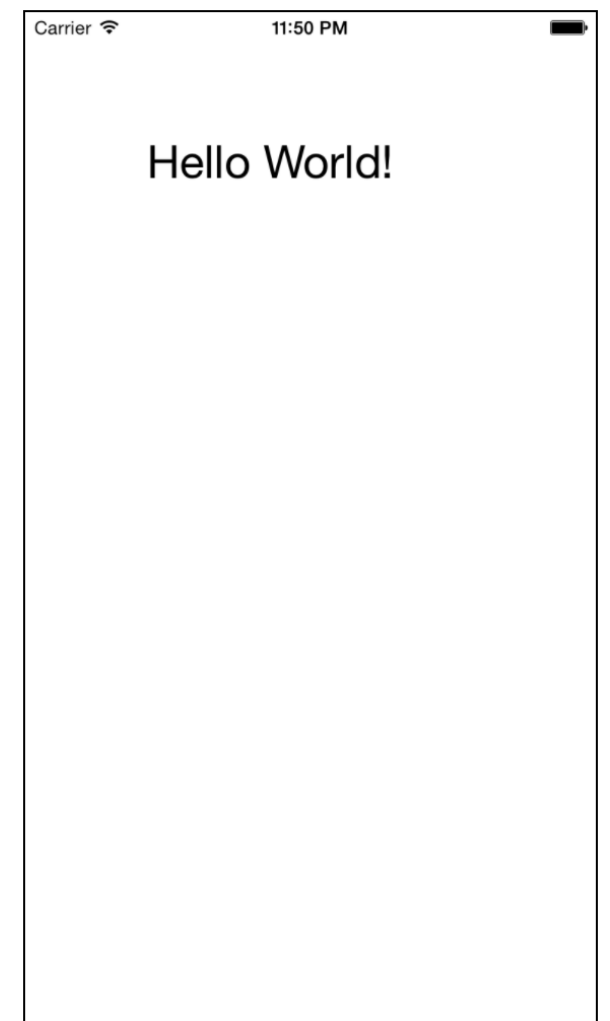
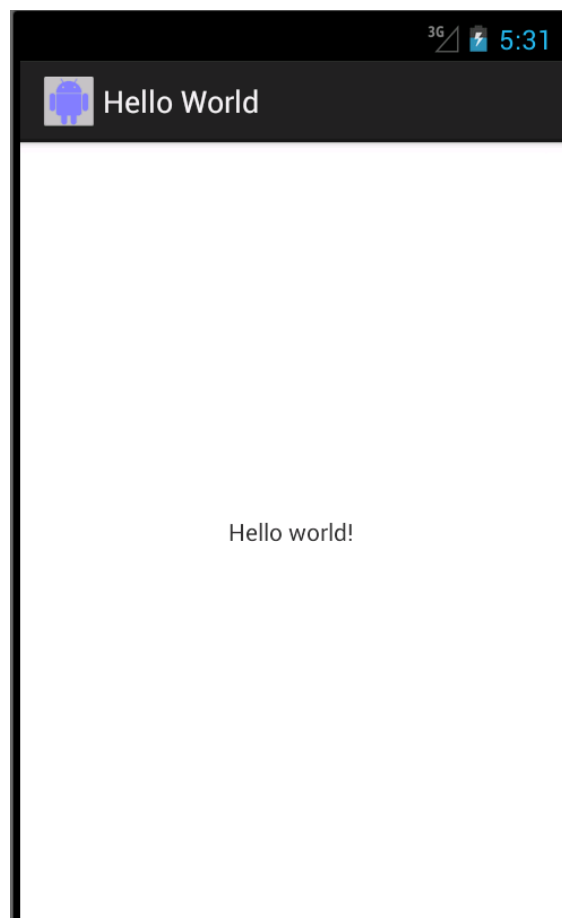
Android - cria uma TextView

iOS - cria uma UILabel

Híbrida-Nativa

Cross compiler (exemplo em Flutter)

```
Widget build(BuildContext context) {  
  return MaterialApp(  
    title: 'Hello World',  
    home: Scaffold(  
      body: Center(  
        child: Text('Hello World'),  
      )),  
  );  
}
```



Esta transformação é feita em
tempo de compilação
(compile-time)

Android e iOS - cria um Canvas nativo no qual
desenha uma label

Híbrida-Nativa

JS Engine

- Como a criação do UI é feita durante a execução, poderá ficar mais lenta. Uma vez criado, é similar a uma aplicação nativa
- Permite atualizar a app sem passar pela App store / Play store (code push)
- Desenvolvimento rápido através do mecanismo de live reload
- O binário (apk, ipa) ocupa mais espaço pois inclui o JS Engine

Cross compiler

- Tudo é feito em compile-time logo o desempenho é similar a uma aplicação nativa
- Atualizações obrigam a publicar uma nova versão da aplicação na respectiva store

Híbrida-Nativa

JS Engine

- Como a criação do UI é feita durante a execução, poderá ficar mais lenta. Uma vez criado, é similar a uma aplicação nativa
- Permite atualizar a app sem passar pela App store / Play store (code push)
- Desenvolvimento rápido através do mecanismo de live reload
- O binário (apk, ipa) ocupa mais espaço pois inclui o JS Engine

Cross compiler

- Tudo é feito em compile-time logo o desempenho é similar a uma aplicação nativa
- Atualizações obrigam a publicar uma nova versão da aplicação na respectiva store

Desvantagem comum: Dependência de uma framework não oficial

Híbrida-Nativa

Algumas frameworks têm uma abordagem mista:

- Desenvolvimento/develop - Usam um Runtime Engine (equivalente ao JS Engine)
 - Isto permite o famoso “Hot Reload”
- Produção/release - Usam o Cross-Compiler
 - Para evitar problemas de desempenho

Flutter e .NET MAUI têm esta abordagem

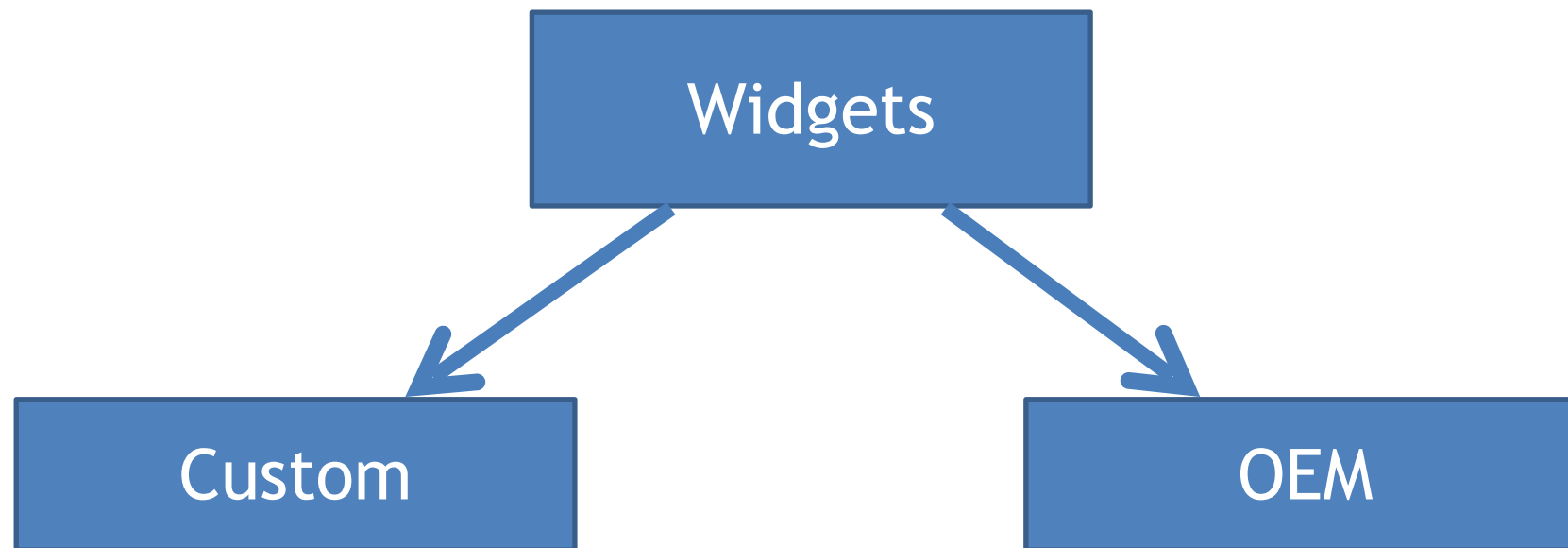
Híbrida-Nativa

Algumas frameworks têm uma abordagem mista:

- Desenvolvimento/develop - Usam um Runtime Engine (equivalente ao JS Engine)
 - Isto permite o famoso “Hot Reload”
- Produção/release - Usam o Cross-Compiler
 - Para evitar problemas de desempenho

Devantagem: Não permite atualizações sem passar pela store

Híbrida-Nativa



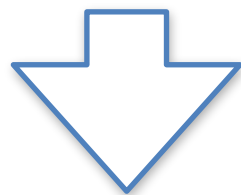
Híbrida-Nativa

Custom Widgets



Flutter

```
ElevatedButton(  
  child: Text("click me"),  
)
```



Android/Kotlin

```
Canvas(  
  drawBitmap(...)
```

(cria um botão virtual mas desenha-o através do sistema operativo)

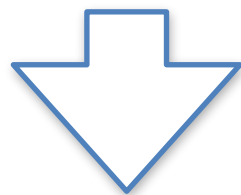
Híbrida-Nativa

Custom Widgets



Flutter

```
ElevatedButton(  
  child: Text("click me"),  
)
```

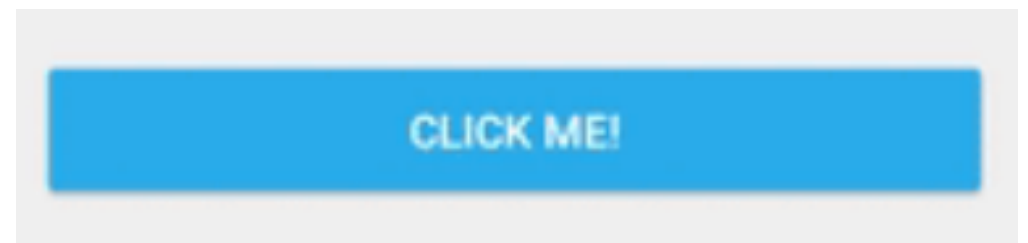


Android/Kotlin

```
Canvas()  
  drawBitmap(...)
```

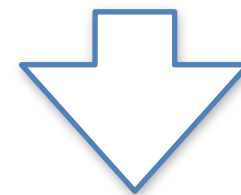
(cria um botão virtual mas desenha-o através do sistema operativo)

OEM Widgets



React-Native

```
<Button title="click me" />
```



Android/Kotlin

```
Button("click me")
```

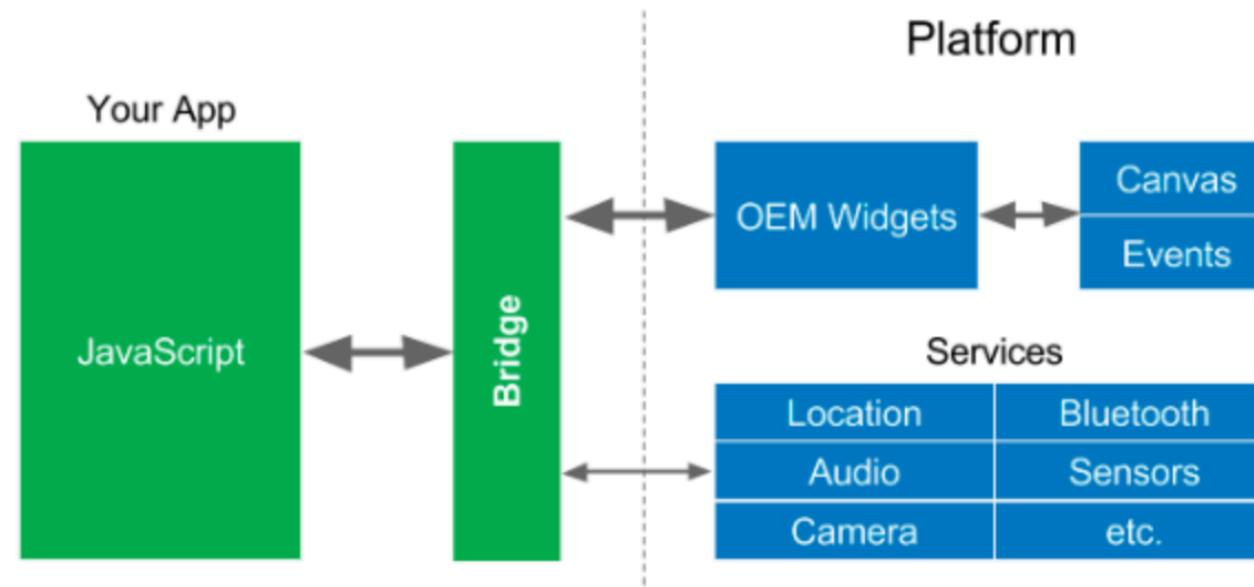
(cria um botão real através da sistema operativo)

Híbrida-Nativa

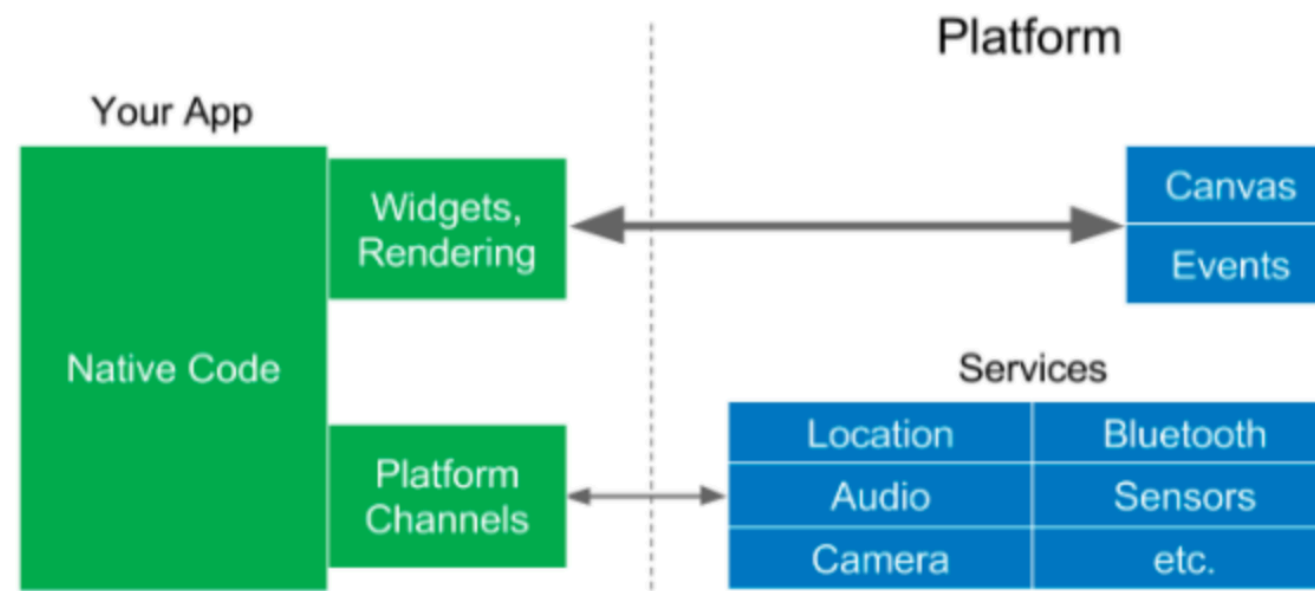
Tabela comparativa

Custom Widgets	OEM Widgets
Componentes gráficos desenhados “à mão” pelas frameworks	Componentes gráficos nativos do SO
Pode parecer estranho pois não usa os componentes que o utilizador está habituado	Segue as guidelines oficiais da interface de cada SO
A aparência é a mesma nos vários SOs	A aparência é necessariamente diferente nos vários SOs
Requer (alguns) conhecimentos de design	Não requer conhecimentos de design
Permite que a app tenha um aspecto moderno mesmo em versões antigas do SO	App está limitada ao aspecto da versão do SO instalado
Flutter	React Native / .NET MAUI

React Native



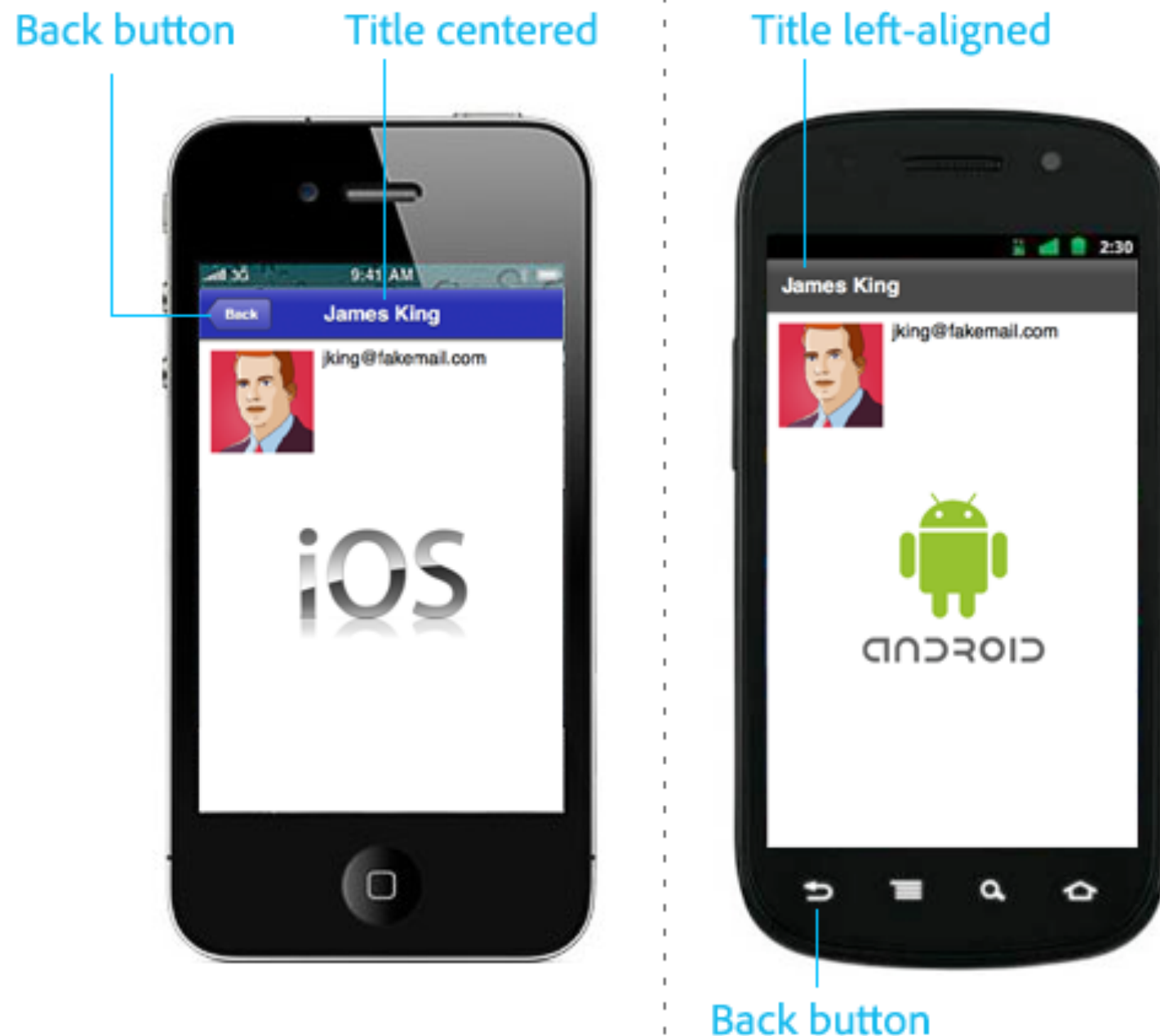
Flutter



Híbrida-Nativa

Problema da interface única

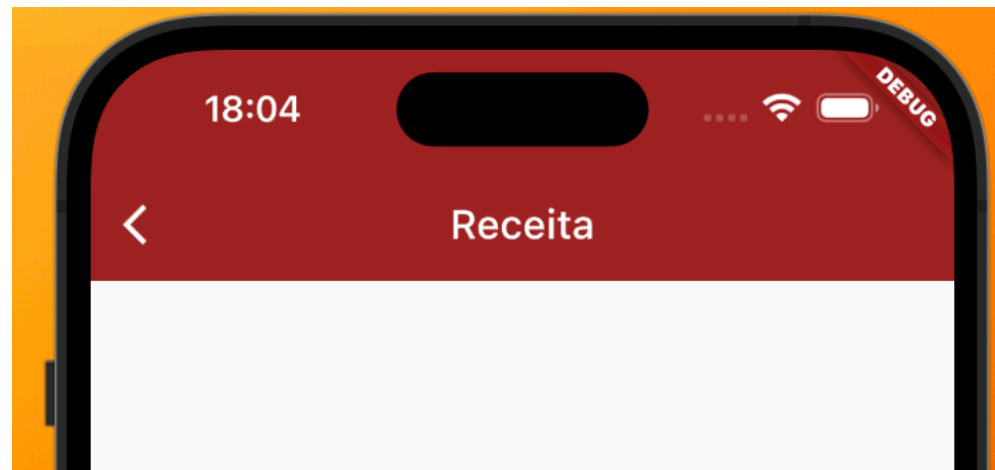
Com a opção Híbrida-Nativa, conseguem-se écrans com aspecto nativo. Mas isso traz outros problemas:



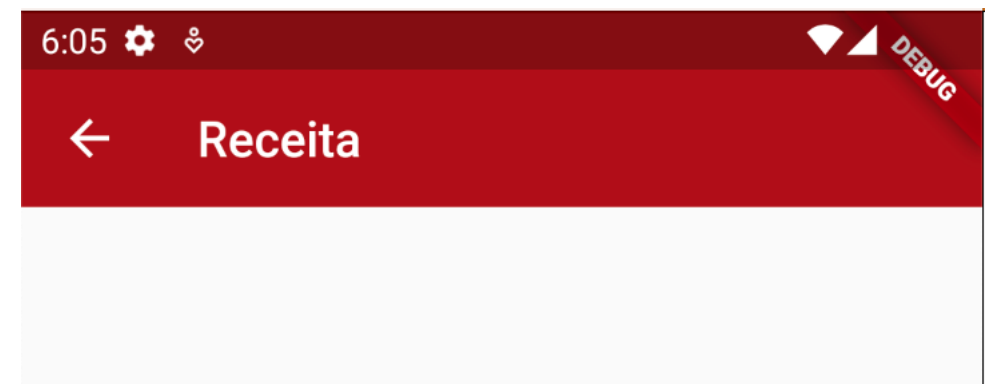
Híbrida-Nativa

Problema da interface única

iOS



Android



Por exemplo, em Flutter, existe um widget AppBar que é desenhado de forma diferente consoante o SO (iOS, Android)

Híbrida-Nativa

Para garantir uma aparência coerente com as regras do respectivo SO, acaba-se por desenvolver a interface gráfica de forma específica, aumentando o esforço de desenvolvimento (que supostamente seria uma das vantagens do modelo híbrido-nativo relativamente ao nativo puro)

Híbrida-Nativa

Para garantir uma aparência coerente com as regras do respectivo SO, acaba-se por desenvolver a interface gráfica de forma específica, aumentando o esforço de desenvolvimento (que supostamente seria uma das vantagens do modelo híbrido-nativo relativamente ao nativo puro)

O mote do React Native:

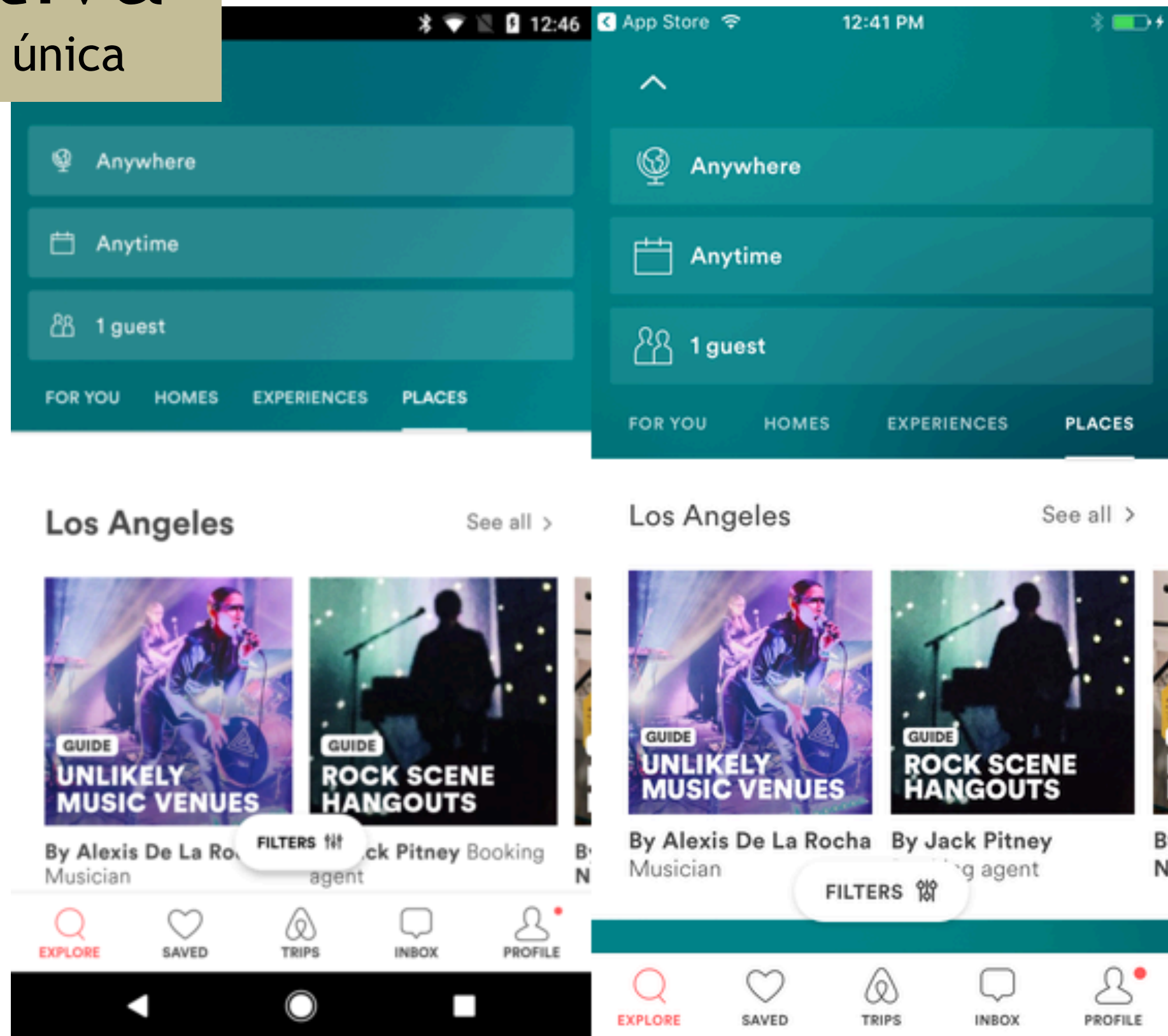
“Learn once, write anywhere”

Híbrida-Nativa

Problema da interface única

Mas será
realmente
um
problema?

(apps nativas Airbnb)



Híbrida-Nativa

Principais frameworks

Framework	Ano início	Empresa	Linguagem	Técnica utilizada	Widgets	Code Push?
Titanium	2008	Axway	Javascript	JS Engine	OEM	Não
React Native	2015	Facebook	Javascript	JS Engine	OEM	Sim
Flutter	2015	Google	Dart	Dart Engine / Cross-compiler	Custom	Sim (**)
.NET MAUI	2022 (*)	Microsoft	C#	Cross-compiler	OEM	Não
Kotlin Multiplatform (KMP)	2023	Jetbrains	Kotlin	Cross-compiler	OEM	Não

(*) Antes disso, a Microsoft oferecia o Xamarin, descontinuado em 2024

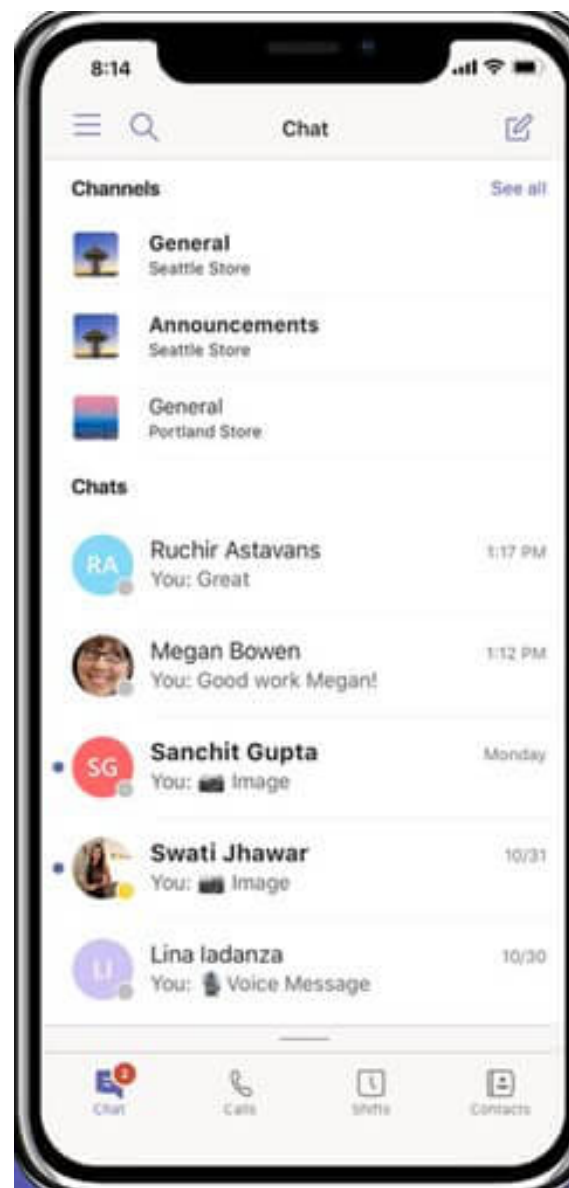
(**) Utilizando shorebird.dev (pago)

Apps desenvolvidas em React Native

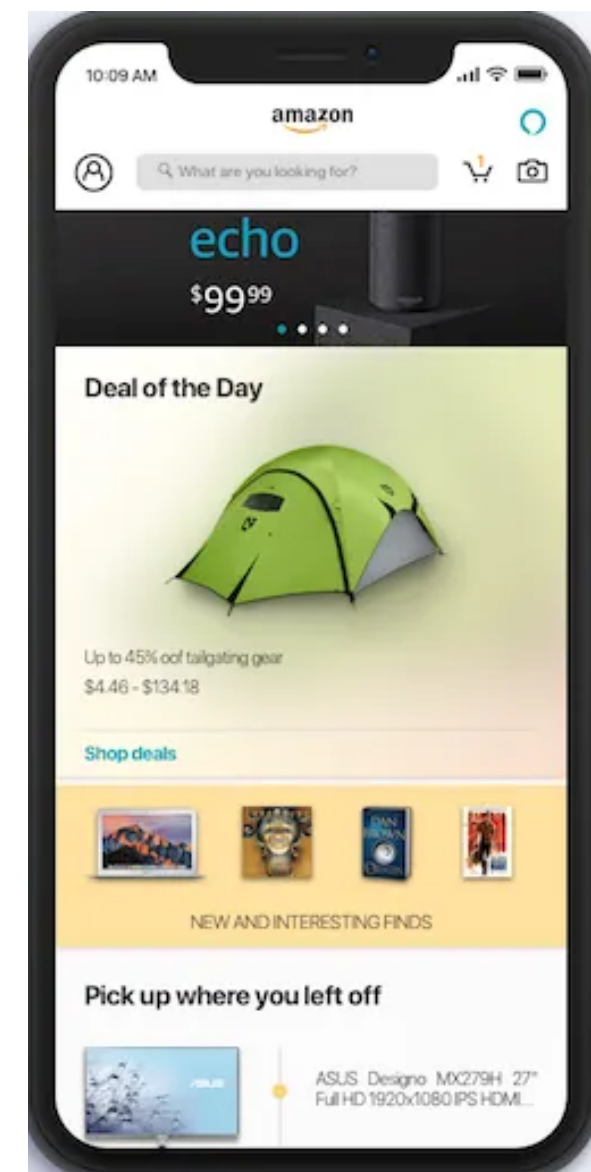
Facebook



Microsoft Teams



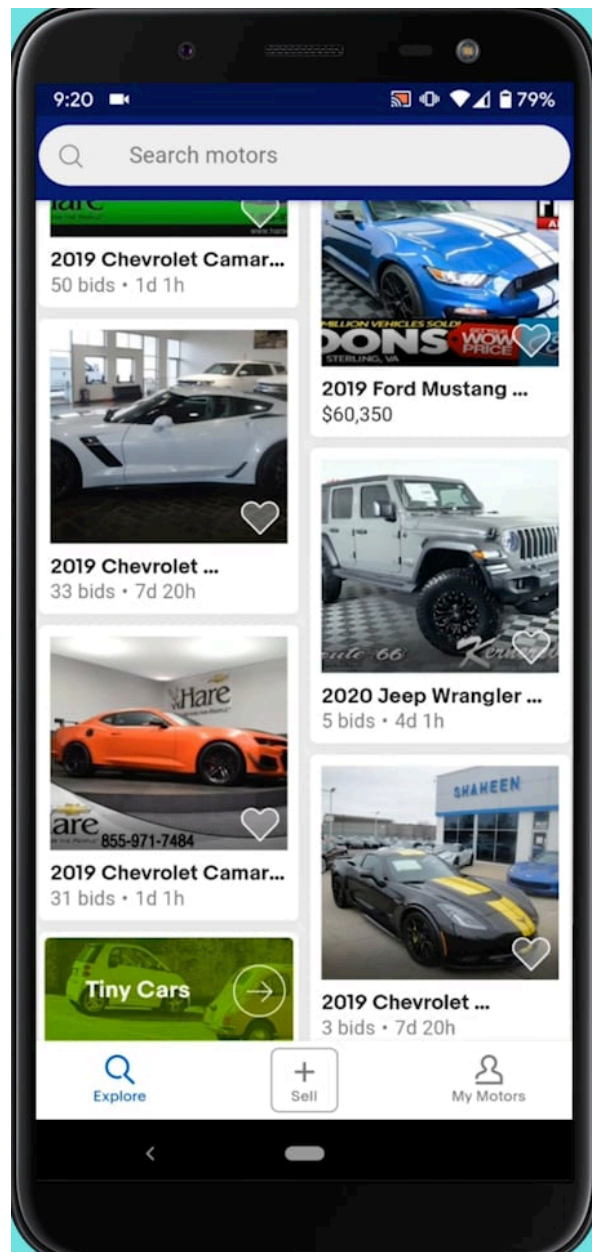
Amazon



<https://reactnative.dev/showcase>

Apps desenvolvidas em Flutter

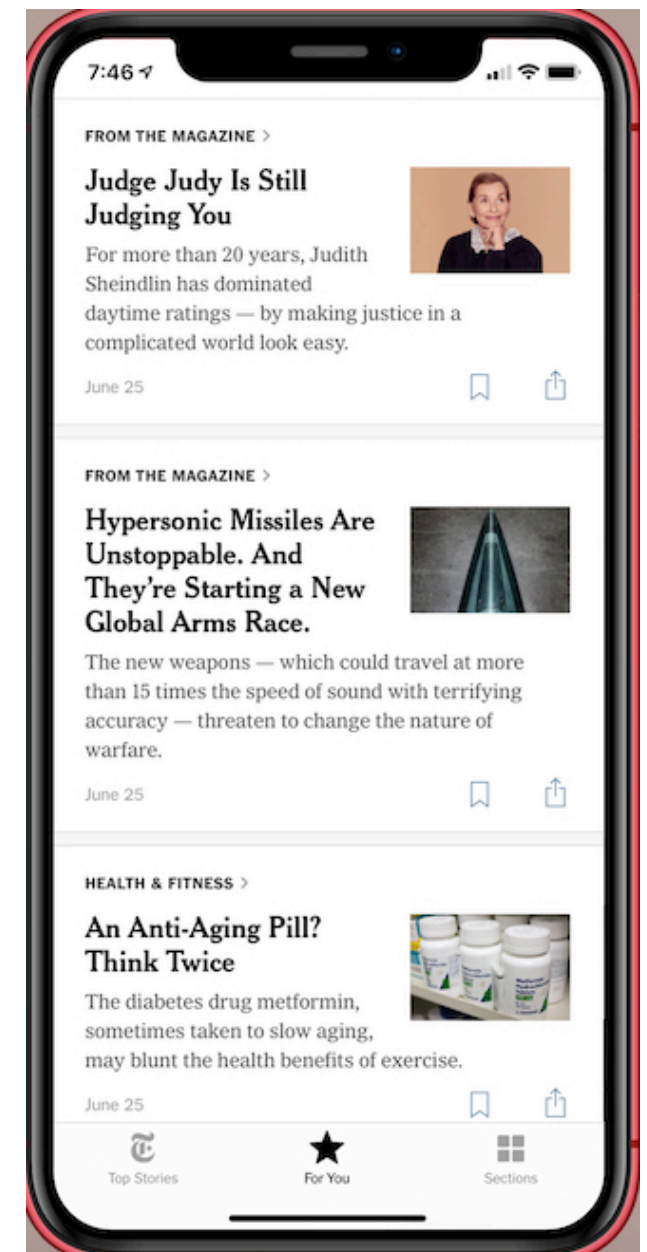
Ebay Motors



Google Earth

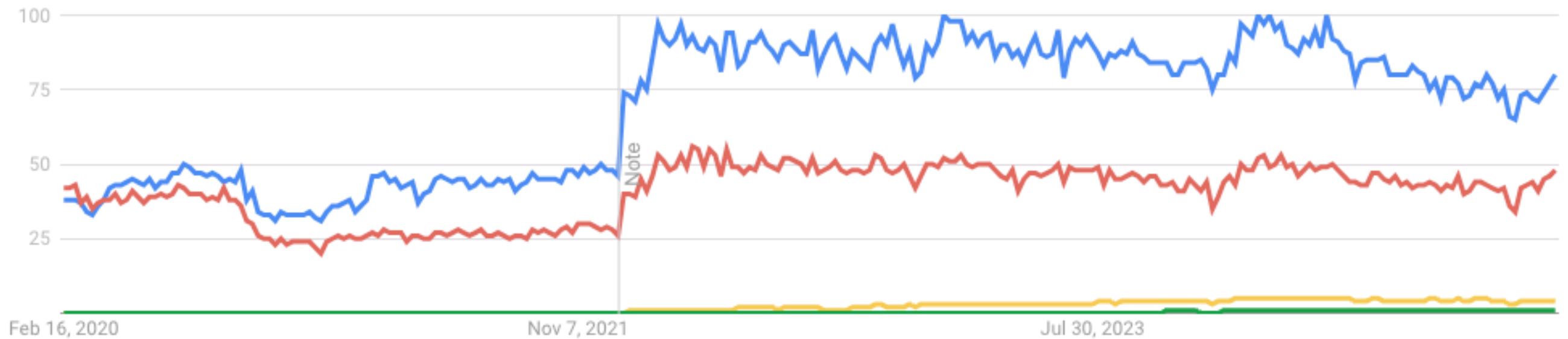


New York Times



Tendência plataformas híbrida-nativa

Pesquisas no Google, desde 2020



Flutter
Software

React Native
Software

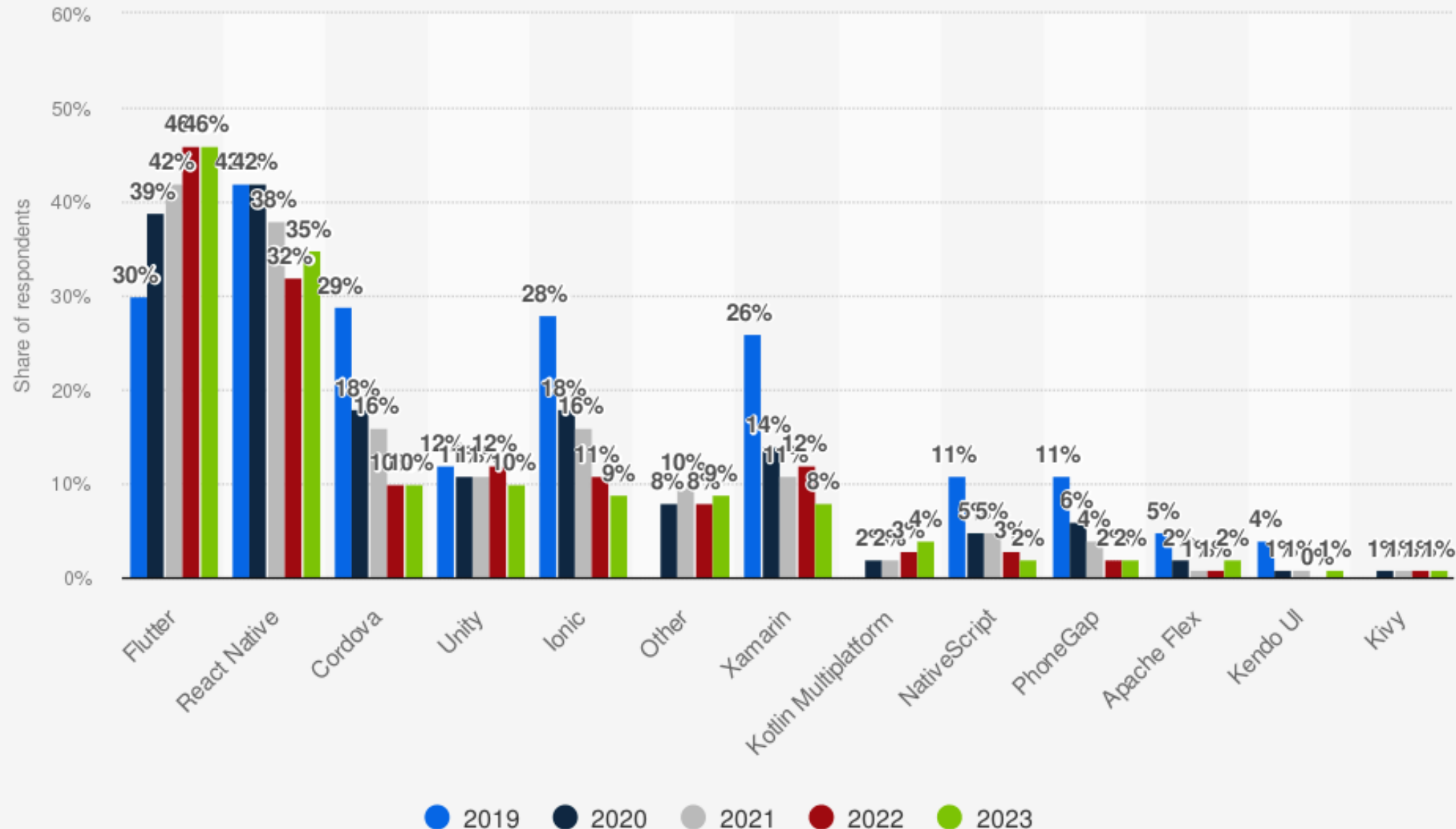
.NET Multi-platfor...
Software

Kotlin Multiplatform
Search term

https://trends.google.com/trends/explore?date=today%205-y&q=%2Fg%2F11f03_rzbg,%2Fg%2F11h03gfy9,%2Fg%2F11mqkqmrcr,Kotlin%20Multiplatform&hl=en

Comparação plataformas híbridas

Cross-platform mobile frameworks used by software developers worldwide
from 2019 to 2023



Source
JetBrains
© Statista 2024

Additional Information:
Worldwide; 2019 to 2023; 29,269 respondents; software developers