

# COMPUTAÇÃO PARA DISPOSITIVOS MÓVEIS

Prof. Jean Paul

**Welcome to the Jungle!**



- Desenvolvedor a mais de 30 anos

## **SOBRE O PROFESSOR**

jean.paul@uniritter.edu.br

Pai, eu sou mau por dentro?  
Não, mas as camadas de  
maldade na superfície vão  
quase até o centro.

 PENSADOR

Homer Simpson



## E QUEM SÃO VOCÊS?

A hora das apresentações

Acessem

<https://socrative.com>

Sala **UNIRITTERJEANPAUL**



## O QUE VAMOS USAR?

Nossas ferramentas

- **Android Studio**  
(<https://developer.android.com/?hl=pt-br>)
- **Android SDK**
- **Linguagem Java**

# Mas podemos usar

- Kotlin
- Objective C (iOS)
- Flutter
- React Native
- WebApps / PWA
- App Inventor (<https://appinventor.mit.edu/>)

Você lembra  
disto?  
Ou pelo  
menos sabe o  
que é?

- O conceito (e a prática) de MVC?
- O que é um objeto?
- O que é uma classe?
- Está familiarizado com HTML/CSS?
- E com Java?



# DEFINIÇÕES E CARACTERÍSTICAS



### Dispositivo Móvel

é um dispositivo de computação pequeno o suficiente para manter e operar na mão tem uma interface de tela, sensível ao toque com botões digitais e teclado ou botões físicos, juntamente com um teclado físico pode se conectar à Internet e interconectar-se a outros dispositivos pode executar sistemas operacionais móveis que permitem aplicativos de terceiros especializados para que esses recursos sejam instalados e executados

### Características

- PORTABILIDADE - ser facilmente deslocado
- USABILIDADE - ser utilizado pelas pessoas em diversos ambientes
- FUNCIONALIDADE - oferecer serviços ao usuário através das aplicações
- TAMANHO REDUZIDO - relacionada com a possibilidade de deslocamento
- CONECTIVIDADE - comunicar com outros sistemas e dispositivos, transmitindo ou recebendo informações

# Dispositivos Móveis Existentes

- Notebooks / Laptops
- Smartphones / Tablets
- Wearables - Smartwatches/ VR's
- Handheld Consoles
- Portable Media Players
- Câmeras Digitais

# Smartphones

Nova era na evolução da computação e dos telefones

Possuem sistema operacional ou plataforma

Em seu início (Ericsson R380 em 2000 - usava o Symbian OS):

- não possuíam tela touch
- tinham poucas funcionalidades
- tinham pouco acesso a diferentes aplicações

Plataformas em 2000: Symbian, Palm Windows Mobile e Blackberry

### Smartphones

Em 2007 com o iPhone, tudo mudou - iniciou a “era touch”

Em 2020, as plataformas Android e iOS dominam o mercado de smartphones estando presentes em mais de 99% dos dispositivos vendidos.



### Smartphones

O Brasil tem hoje dois dispositivos digitais por habitante, incluindo smartphones, computadores, notebooks e tablets.

Em dez/2020 o Brasil tinha 234,1 milhões de celulares, uma densidade de 110,07 cel/100 hab.

Só em dez/2020 2 milhões de novos celulares foram ativados, destes 391 mil são pré-pagos.

Fonte: <https://www.teleco.com.br/ncel.asp>

# Smartphones

Milhares	Dez/19	Nov/20	Dez/20
Celulares	226.674	232.055	234.067
Pré-pago	51,64%	49,42%	49,16%
Densidade*	107,29	109,18	110,07
Crescimento Mês	-2.138	1.826	2.012
	-0,9%	0,8%	0,9%
Crescimento Ano	-2.528	5.382	7.394
	-1,1%	2,4%	3,3%
Crescimento em 1 ano	-2.528	3.244	7.394
	-1,10%	1,4%	3,3%

Fonte: <https://www.teleco.com.br/ncel.asp>



# PRINCIPAIS PLATAFORMAS



# Android

Plataforma da Google com foco em mobile

Pilha de software baseada no SO Linux

Cada camada da pilha possui várias funções específicas que podemos usar em aplicativos Android

## Principais Plataformas

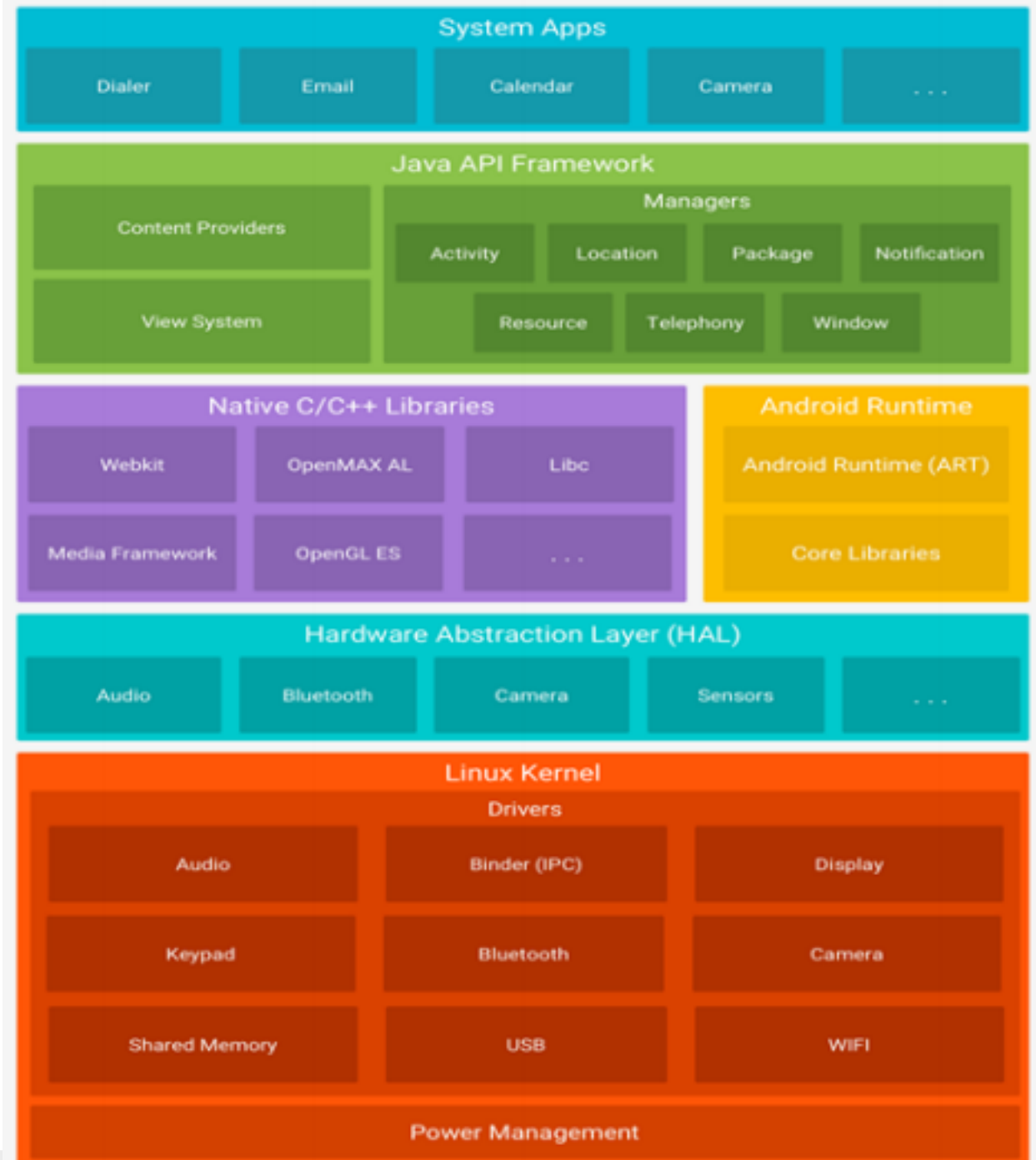
### Android

A base é o kernel, que gerencia a memória, segurança, energia e drives

A camada HAL, bibliotecas específicas para hardware como câmera, bluetooth

A camada ART compila o código da APK para código de máquina

A Java API oferece acesso aos recursos do SO Android como gerenciador de atividades, provedores de conteúdo, etc.



# Sistema Operacional Android

Sistema com o código aberto

Versão mais recente é a 11 (Android11), lançada em set/2020.

Mas o legado é grande (fonte Goolge – mai/2019):

- **Android 9 Pie:** 31,3%
- **Android 8 Oreo:** 21,3%
- **Android 7 Nougat:** 12,9%
- **Android 6 Marshmallow:** 11,2%
- **Android 5 Lollipop:** 9,2%
- **Android 10:** 8,2%
- **Android 4.4 KitKat:** 4%
- **Android 4.1 até 4.3 Jelly Bean:** 1,7%
- **Android 4.0 Ice Cream Sandwich:** 0,2%

# Android Desenvolvimento

Usa as linguagens Java e Kotlin

Oferece um SDK com inúmeras ferramentas (emulador, monitoramento, gerenciamento de builds, etc.)

IDE oficial é o Android Studio

Possui versões para Windows, MAC OS e Linux



### iOS

Plataforma da Apple para desenvolvimento mobile

Abreviatura para iPhone Operation System

Baseada no MAC OS X

Versão Atual iOS14

O iOS 13 está disponível na maior parte dos iPhones em uso lançados até os dias atuais. De acordo com a companhia, o sistema operacional está instalado em 81% dos celulares da marca, acompanhado pelo iOS 12, com 13%. As demais versões, como iOS11, iOS 10 e anteriores, acumulam uma fatia de 6% de distribuição.(fonte Tecnoblog - [\)](https://tecnoblog.net/347187/ios-13-esta-em-81-de-todos-os-iphones-em-uso-afirma-apple/#:~:text=De%20acordo%20com%20a%20companhia,fatia%20de%206%25%20de%20distribui%C3%A7%C3%A3o.)

### iOS

- OS - frameworks e interfaces básicas do sistema para comunicação da app com o hardware do dispositivo
- Services - interfaces para acesso aos serviços nativos da plataforma como arquivos, contatos, SQLite
- Media - componentes para gerenciamento de mídia
- Touch - componentes para a interface do usuário, eventos de tela



### iOS Desenvolvimento

Usa as linguagens Objective-C e Swift

Oferece um SDK com inúmeras ferramentas (emulador, monitoramento, gerenciamento de builds, etc.)

IDE oficial é o XCode

Possui versões para MAC OS



# FERRAMENTAS DE DESENVOLVIMENTO



# Abordagens de desenvolvimento

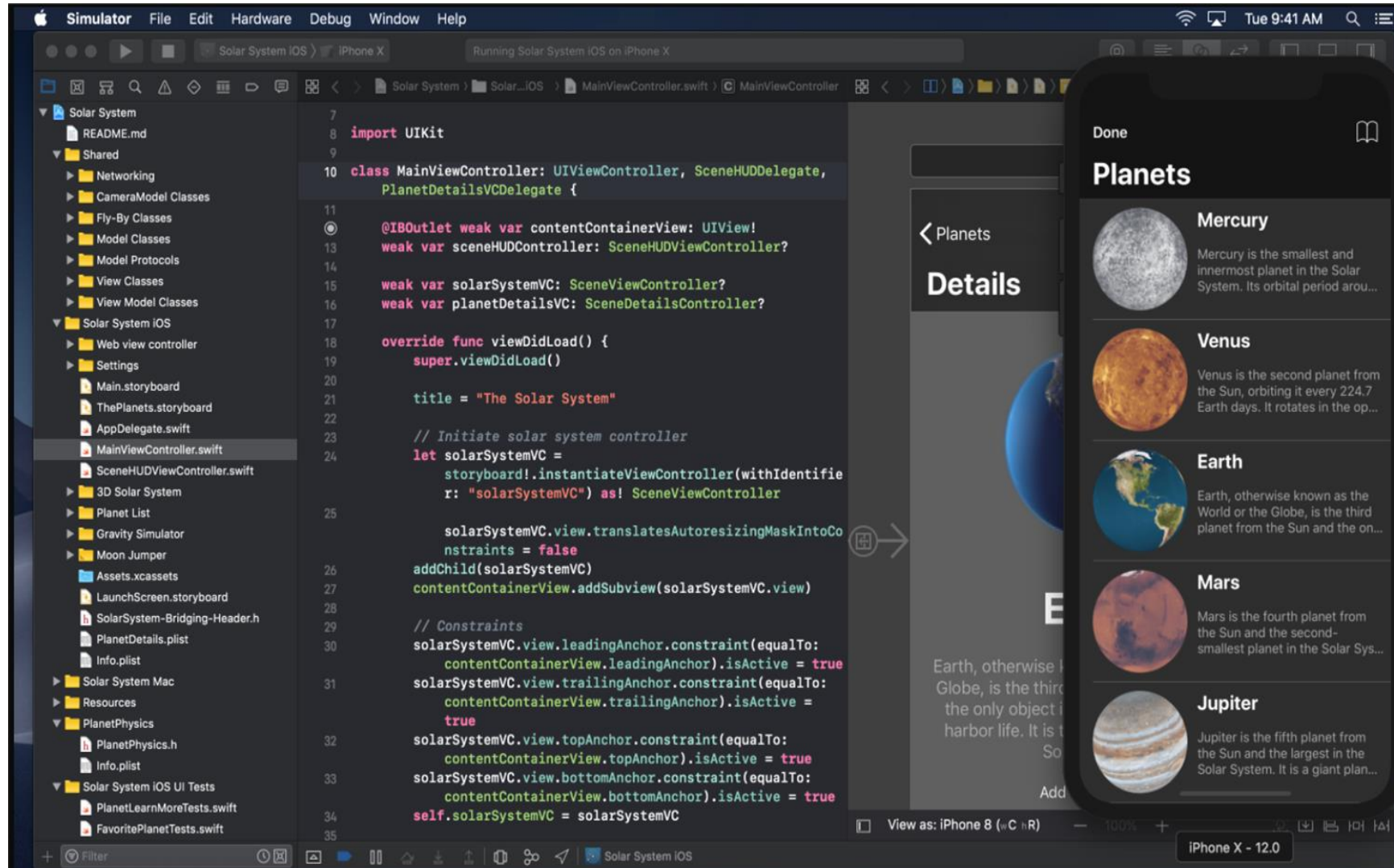
- Abordagem nativa
- Abordagem nativa cross-platform
- Abordagem web
- Abordagem híbrida

### Abordagem nativa

- Usa o SDK disponibilizado
- Melhor experiência do usuário
- Necessidade de uso massivo de recursos da plataforma
- Necessidade de otimização de desempenho
- Necessidade de publicação local e em lojas oficiais das plataformas

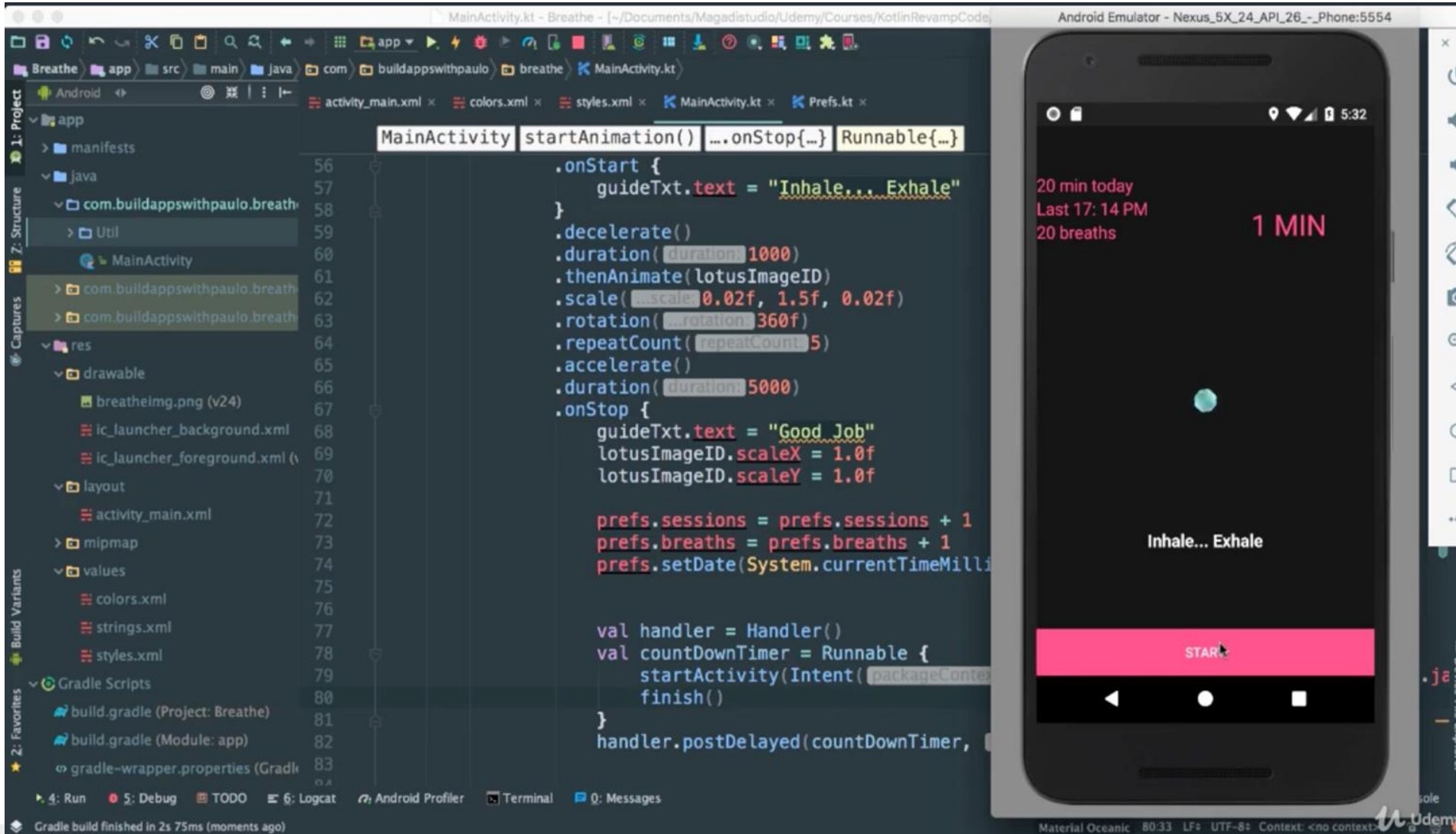
# Ferramentas de Desenvolvimento – Abordagem Nativa

## XCode



# Ferramentas de Desenvolvimento – Abordagem Nativa

## Android Studio



## Ferramentas de Desenvolvimento – Abordagem Nativa Cross-Plataform

- Desenvolve para inúmeras plataformas usando frameworks
- Converte código próprio em código nativo
- Unifica diferentes linguagens e recursos
- Desempenho pode ser prejudicado
- Sensível às mudanças em cada plataforma

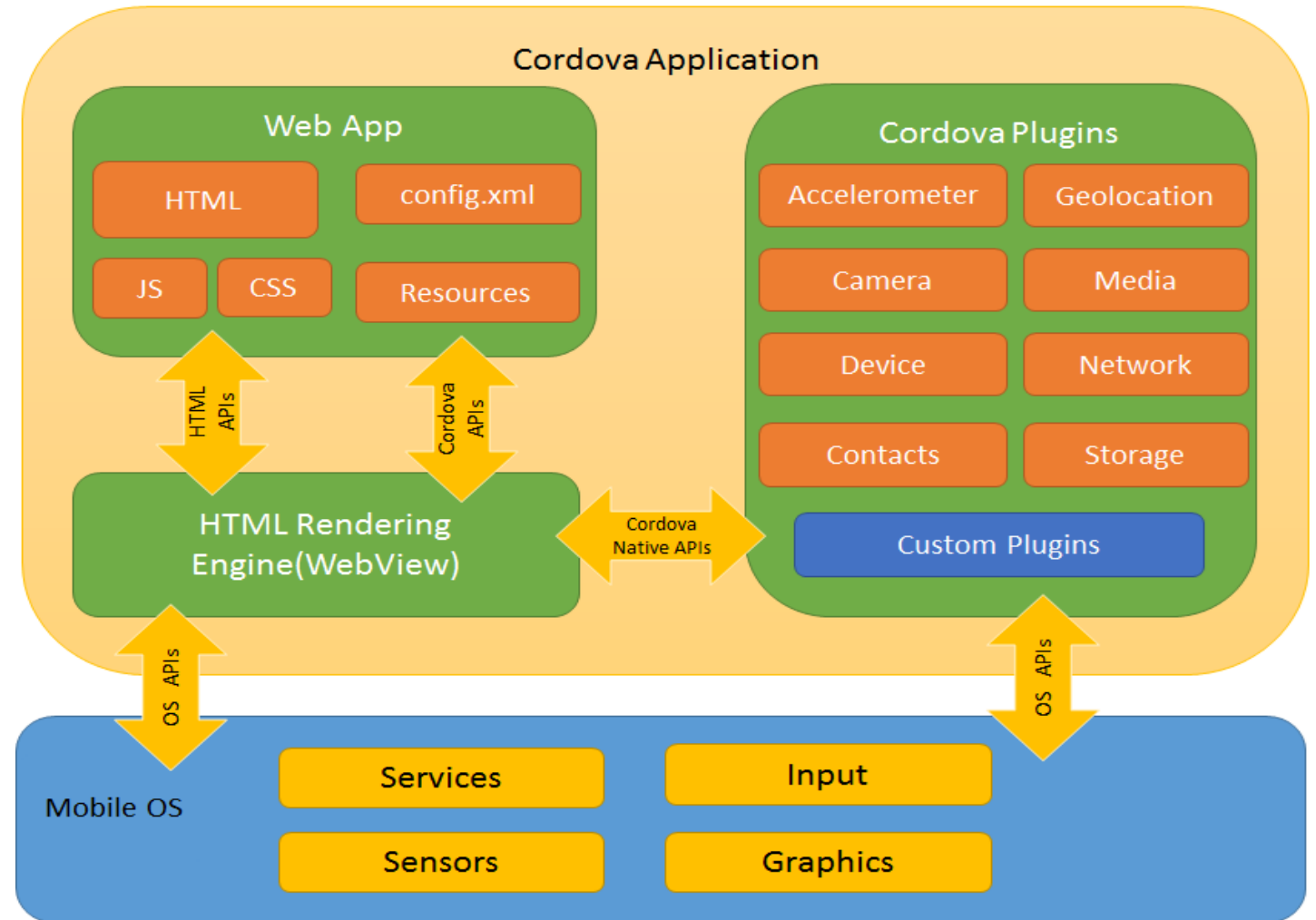
# Ferramentas de Desenvolvimento – Abordagem Nativa Cross-Plataform

## Apache Cordova

Framework de desenvolvimento móvel de código aberto

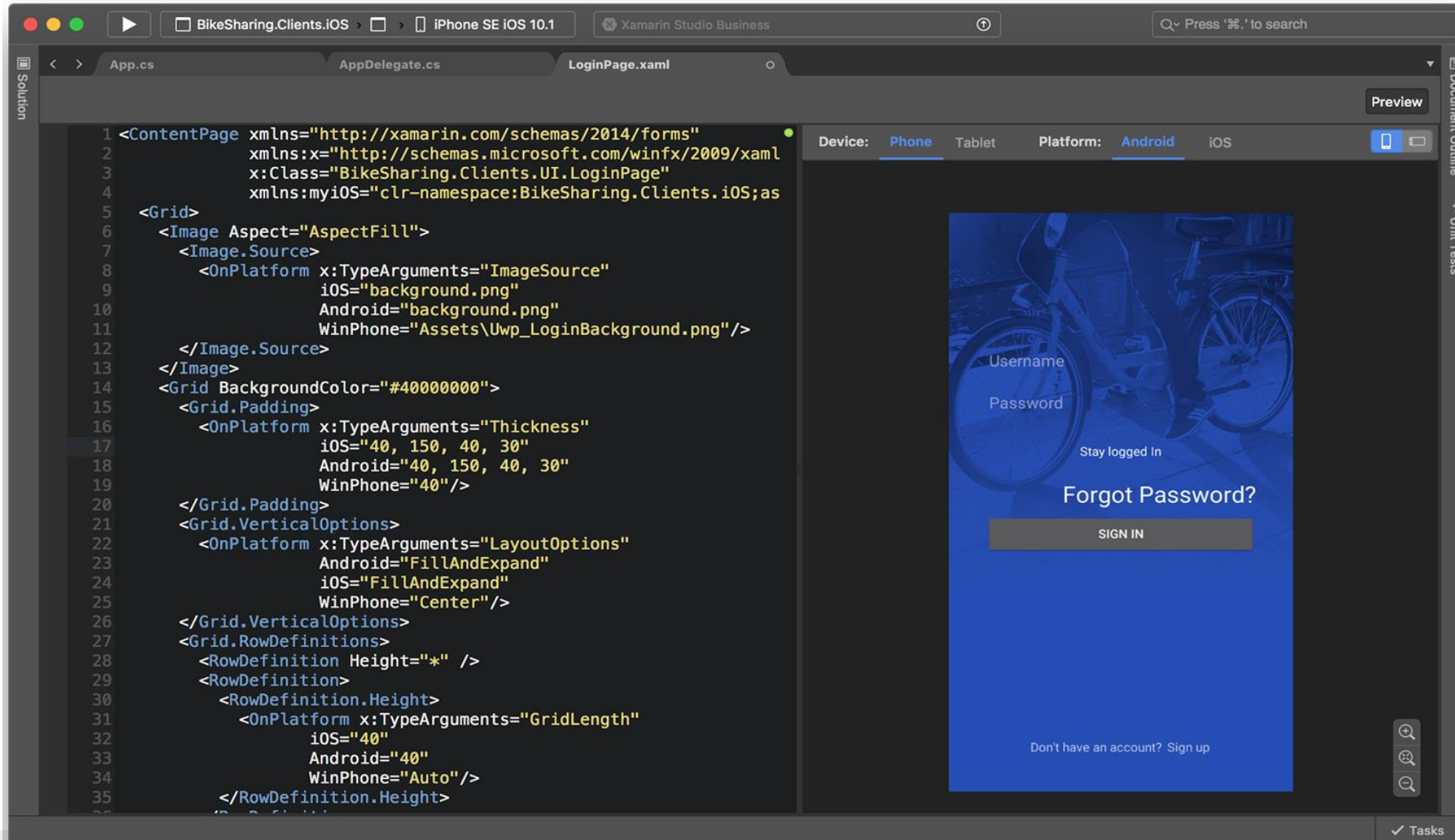
Permite que o uso de tecnologias padrão da web - HTML5, CSS3 e JavaScript para desenvolvimento em várias plataformas

Os aplicativos são executados em wrappers destinados a cada plataforma

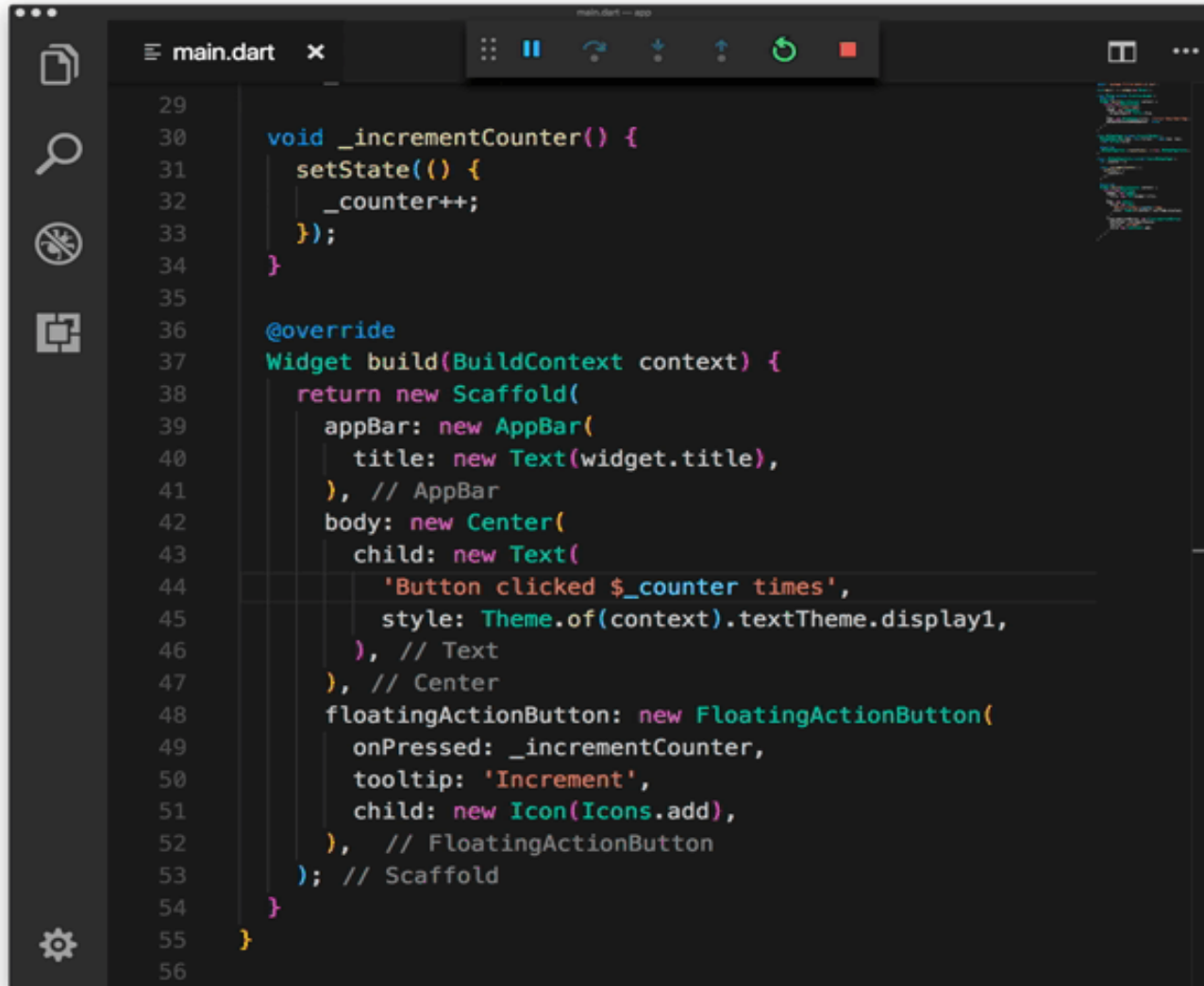




## Xamarin Studio

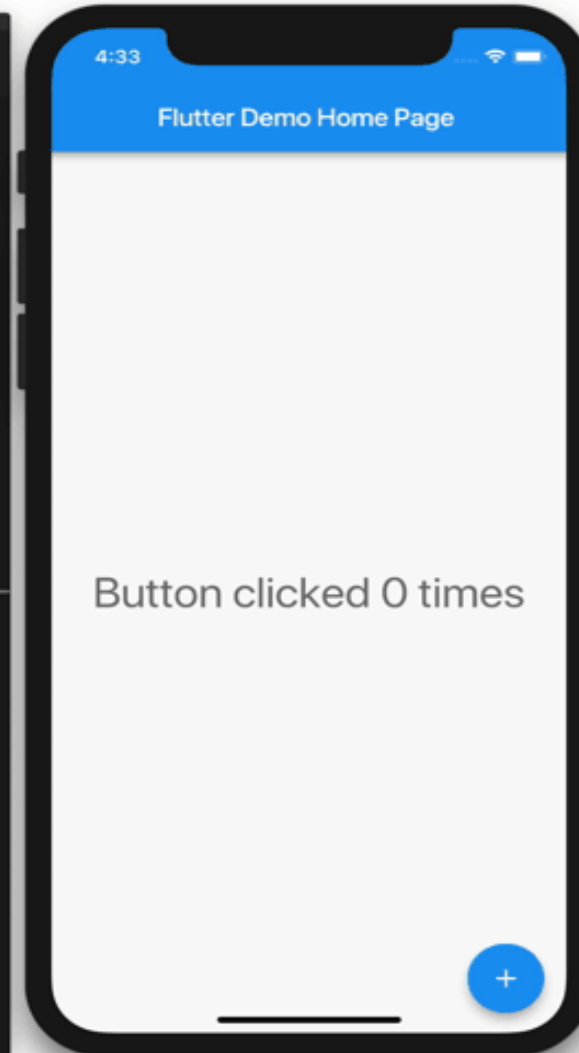


## React Native



The image shows a code editor window with a dark theme. The file name is 'main.dart'. The code is in Dart and implements a simple counter app. It includes a method to increment the counter and a build method that creates a Scaffold with an AppBar, a Center widget containing a Text widget, and a FloatingActionButton with an 'Increment' tooltip and an 'add' icon.

```
29
30 void _incrementCounter() {
31   setState(() {
32     _counter++;
33   });
34 }
35
36 @override
37 Widget build(BuildContext context) {
38   return new Scaffold(
39     appBar: new AppBar(
40       title: new Text(widget.title),
41     ), // AppBar
42     body: new Center(
43       child: new Text(
44         'Button clicked $_counter times',
45         style: Theme.of(context).textTheme.display1,
46       ), // Text
47     ), // Center
48     floatingActionButton: new FloatingActionButton(
49       onPressed: _incrementCounter,
50       tooltip: 'Increment',
51       child: new Icon(Icons.add),
52     ), // FloatingActionButton
53   ); // Scaffold
54 }
55
56
```





# Ferramentas de Desenvolvimento - Abordagem Nativa Cross-Plataform

## Kony

The screenshot displays the Kony Visualizer IDE interface, which is used for developing native cross-platform mobile applications. The interface is divided into several panels:

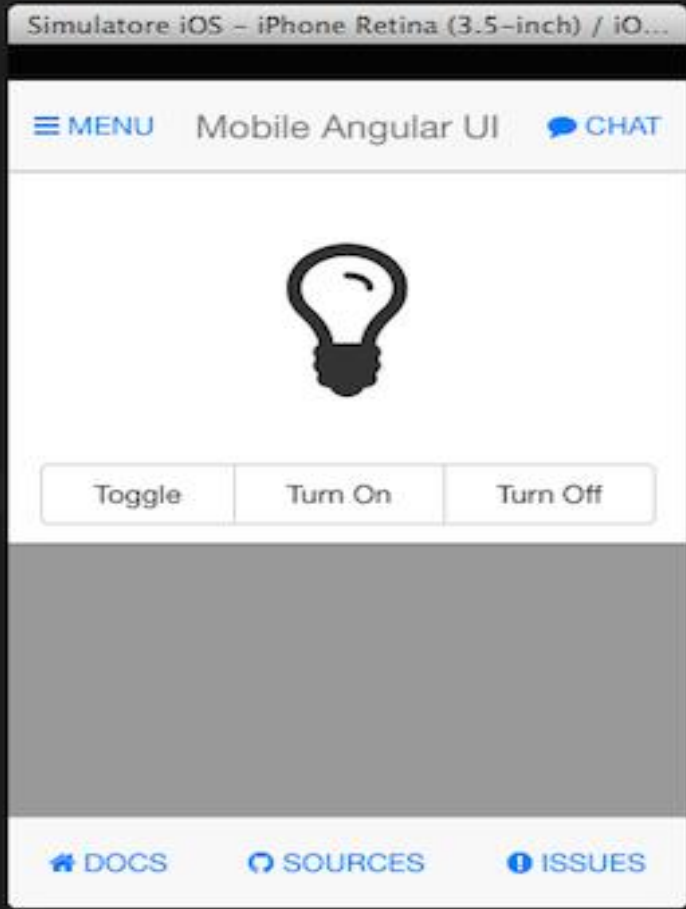
- Project Panel (Left):** Shows the project structure, including components like `masterContainer`, `shadowRight`, `formReturn`, `innerContainer`, `lgContainer`, `masterScroll`, `adPager`, `contentPage1`, `CopyLabel0627b`, `CopyLabel06a0b`, `contentPage2`, `contentPage3`, `contentPage4`, `contentArea`, `activitySegment`, `Label00c267b07aa49`, `contentAreaShadow`, `pagerPages`, `appMenuContainer`, `headerContainer`, `headerWhite`, and `headerTrans`.
- Design Canvas (Center):** Displays a mobile app design for an iPhone 6 Standard (750 x 1334). The app is titled "myHealth" and features a "COVERAGE" section. It includes a bar chart showing "Aetna Covers" for "Jane Williams" (80%) and "Greg Williams" (100%). Below the chart, there are three donut charts representing "In-Network" (80%), "Out-Of-Network" (60%), and "Prescription" (90%). At the bottom, there are sections for "Flexible Spending" (\$1,982), "Balances" (\$200), and "Vision Credit".
- Code Editor (Right):** Shows the JavaScript code for the app. The code includes functions for `frmLoginPreshow()`, `frmLoginPostshow()`, `resetLogin()`, and `loginAction()`. The code uses Kony's animation framework to animate various UI elements.
- Properties Panel (Far Right):** Allows users to configure the properties of the selected widget. The "Normal" tab is active, showing properties like Name, Platform, Background, Border, and Shadow. The "Focus" and "Blocked UI" tabs are also visible.

### Abordagem Web

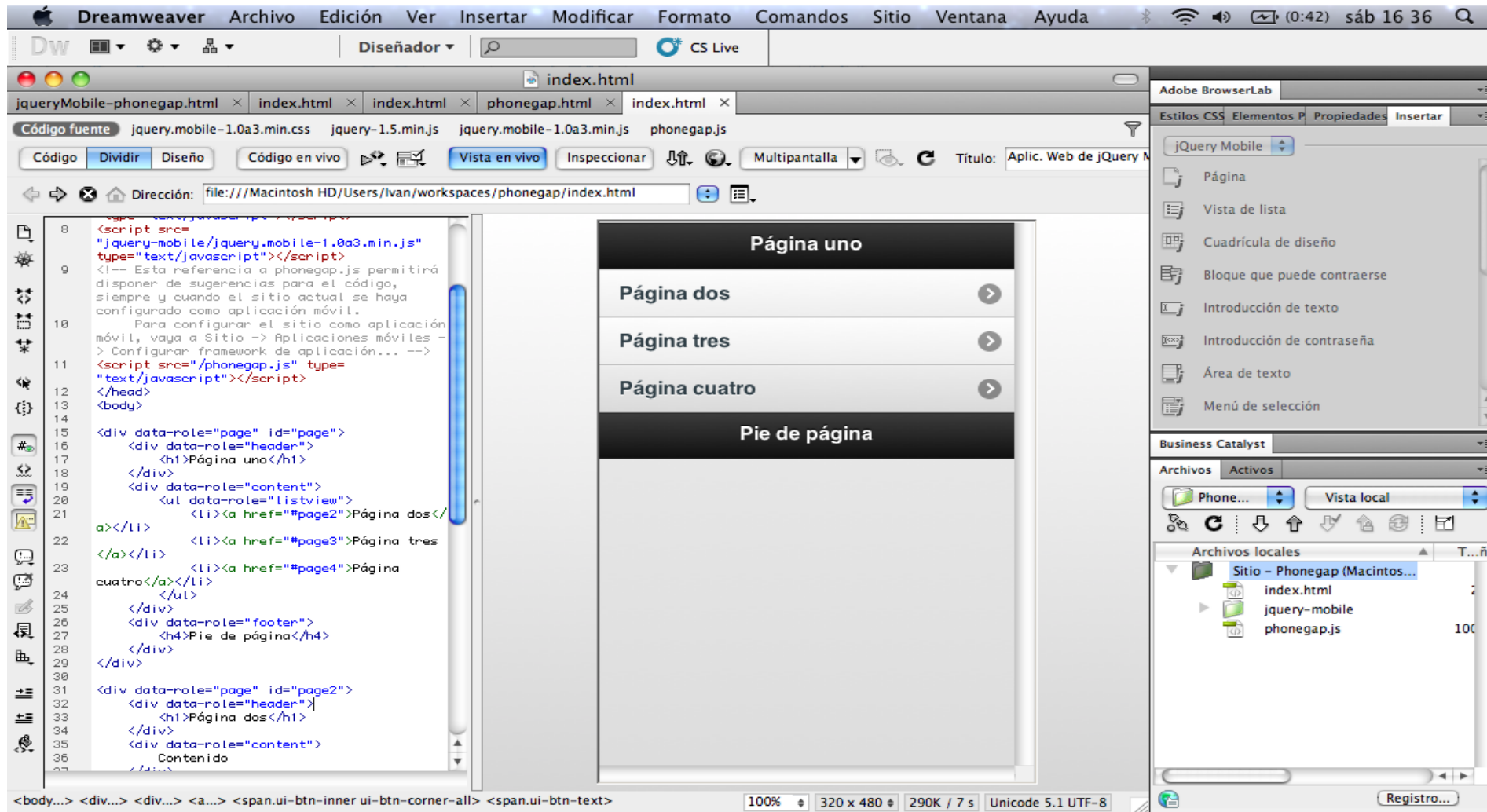
- Usa o navegador web e as tecnologias web (HTML, CSS, JS, etc.) do dispositivo
- Necessidade de diminuir o tempo e custo no desenvolvimento
- Aplicações são hospedadas em servidor e não localmente
- Acesso restrito aos recursos de cada plataforma
- Necessita acesso à *internet*
- Permite acessar todas as plataformas
- Não precisa disponibilizar nas lojas oficiais

## Mobile

```
5 <p toggleable id="lightbulb" active-class="text-primary" class="text-default">
6   <i class="fa fa-lightbulb-o"></i>
7 </p>
8
9 <div class="btn-group justified nav-tabs">
10  <a
11    toggle="toggle"
12    target="lightbulb"
13    active-class="active"
14    class="btn btn-default" href>
15    Toggle
16  </a>
17  <a
18    toggle="on"
19    target="lightbulb"
20    class="btn btn-default" href>
21    Turn On
22  </a>
23  <a
24    toggle="off"
25    target="lightbulb"
26    class="btn btn-default" href>
27    Turn Off
28  </a>
29 </div>
30 </div>
```



## jQuery Mobile



NativeScript

PlaygroundEditor

My Playground

NewQR codeSaveForkDownload

Preview

Explorer

home.component.html

app

app-routing.module.ts

app.component.html

app.component.ts

app.css

app.module.ts

home

home-routing.module.ts

home.component.css

home.component.html

home.component.ts

home.module.ts

main.ts

Components

ESSENTIALS

Button

Label

Image

Slider

Switch

Text View

List View

Progress

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

<ActionBar title="Home" class="action-bar">

</ActionBar>

<GridLayout>

<ScrollView class="page">

<StackLayout class="home-panel">

<!--Add your page content here-->

<Label textWrap="true" text="Play with NativeScript!" class="h2 description-label"></Label>

<Label textWrap="true" text="Write code in the editor or drag and drop components to build a NativeScript mobile ap

class="h2 description-label"></Label>

<Label textWrap="true" text="Scan the QR code with your mobile device and watch the changes sync live while you pla

class="h2 description-label"></Label>

</StackLayout>

</ScrollView>

</GridLayout>

LESSON 1. Getting Your App Up and Running

11:05

Home

NativeScript

Write code in the editor or drag and drop components to build a NativeScript mobile application.

Scan the QR code with your mobile device and watch the changes sync live while you play with the code.

11:02

Home

NativeScript

Write code in the editor or drag and drop components to build a NativeScript mobile application.

Scan the QR code with your mobile device and watch the changes sync live while you play with the code.

You can develop on multiple devices simultaneously. If you have multiple iOS or Android devices, download the NativeScript Playground app on each and scan this app's QR code.

In NativeScript Playground the Ctrl + S or Cmd + S keyboard shortcut saves all your files. Later in this guide you'll be changing multiple files, and you can use that single keyboard shortcut to save all of them at the same time.

One of the biggest benefits of using NativeScript is the fast development workflow you just saw in action. You have the ability to change your code and see those changes instantly—even on multiple devices.

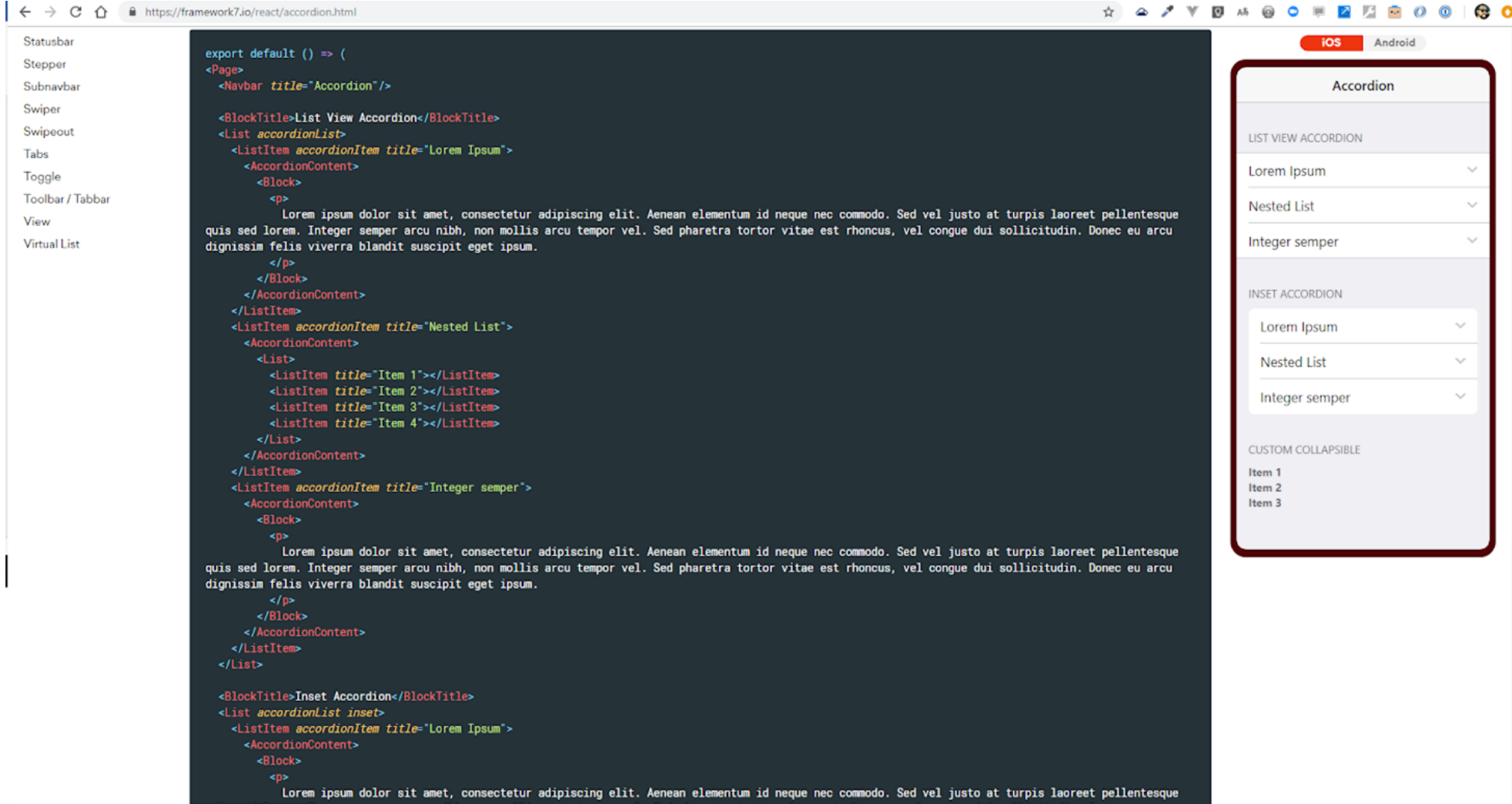
Prev step

2/2

Complete tutorial



# Framework7



### Abordagem Híbrida

- Combinação das abordagens anteriores (nativa + web)
- Usam o motor *web*
- Frameworks encapsulam as chamadas às API's nativas
- Ambiente único de desenvolvimento
- Em tempo de compilação, código é convertido para código nativo
- Podem ser distribuídas localmente ou em lojas oficiais

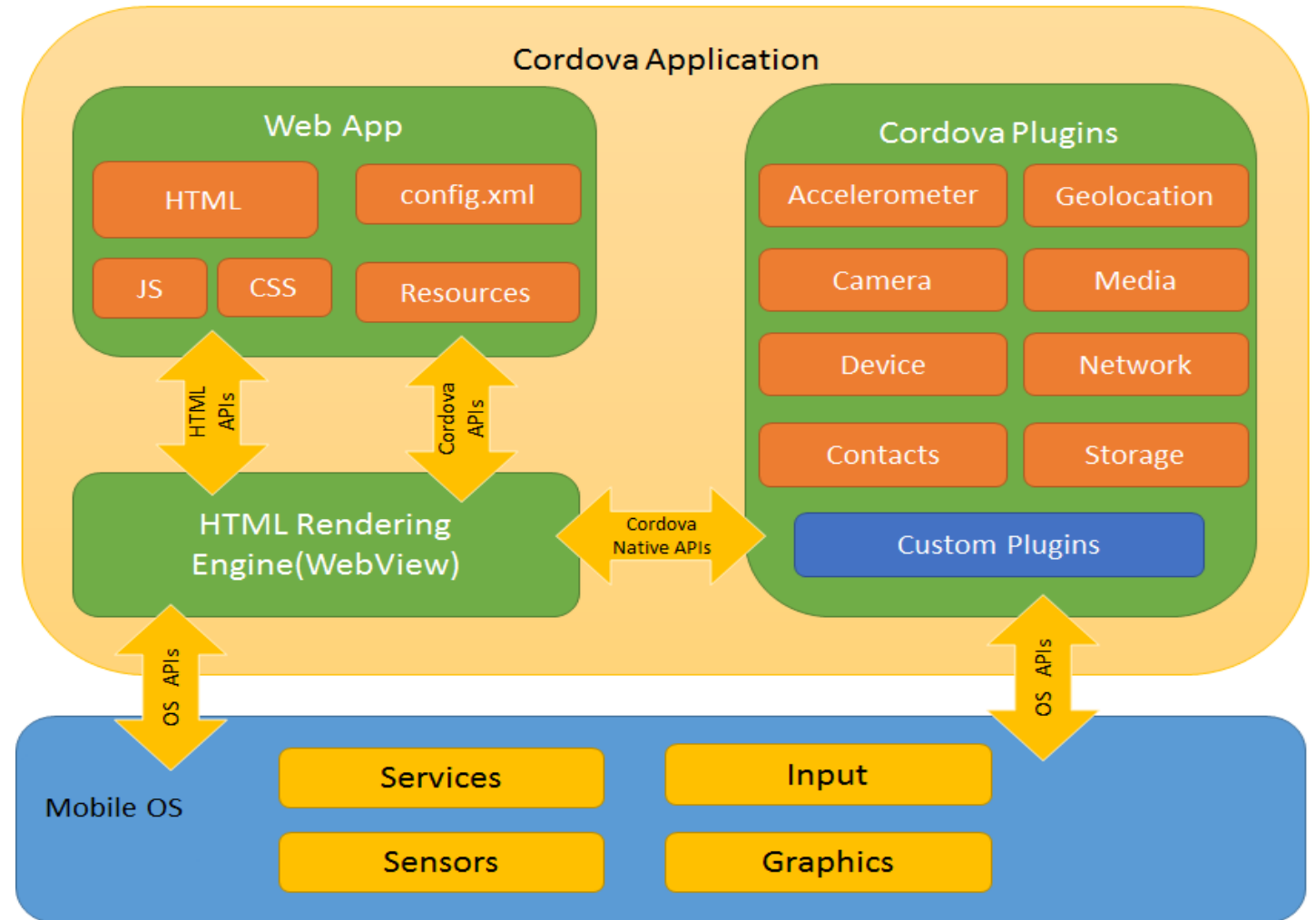
# Ferramentas de Desenvolvimento – Abordagem Híbrida

## Apache Cordova

Framework de desenvolvimento móvel de código aberto

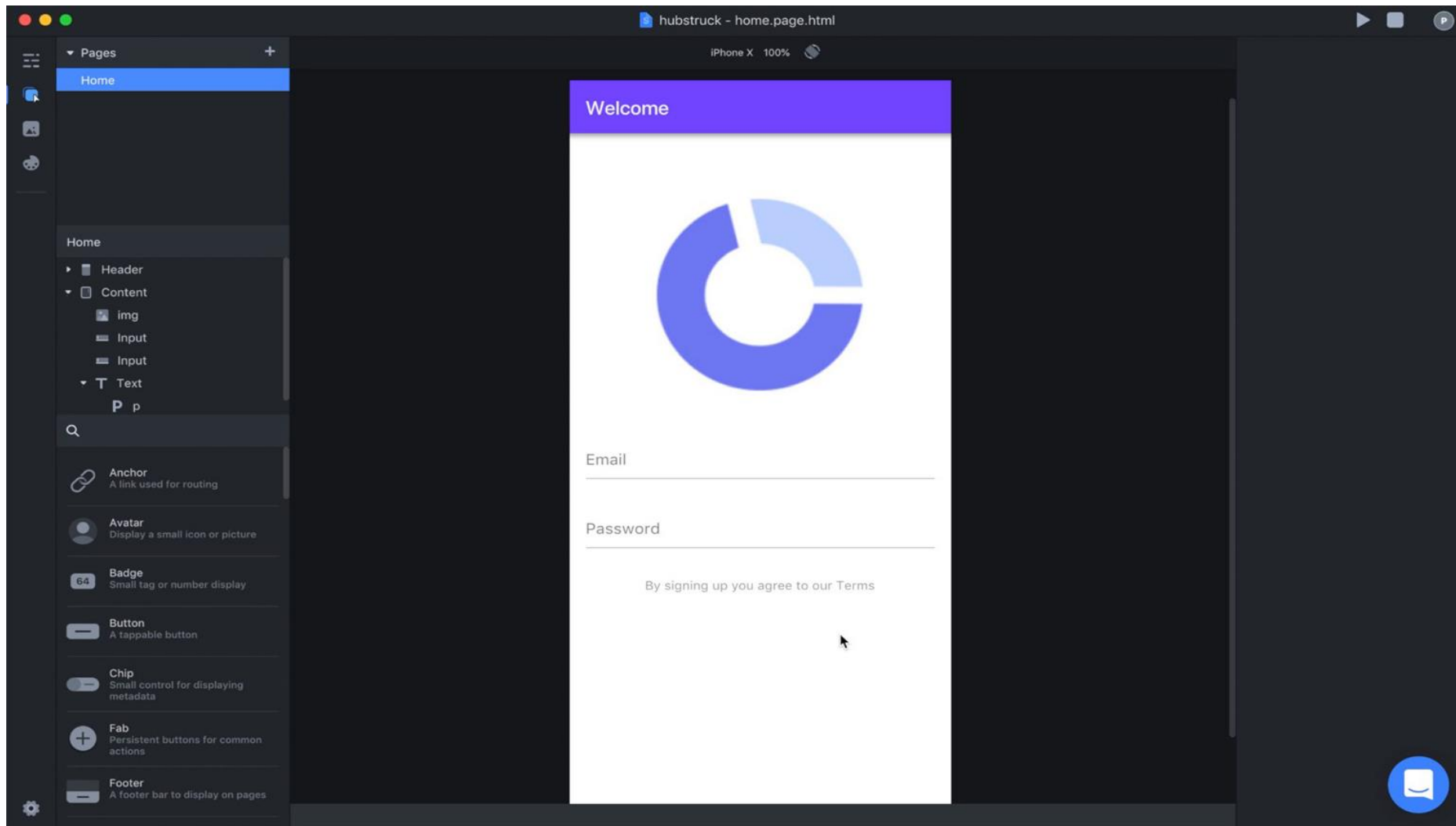
Permite que o uso de tecnologias padrão da web - HTML5, CSS3 e JavaScript para desenvolvimento em várias plataformas

Os aplicativos são executados em wrappers destinados a cada plataforma





## Ionic Studio



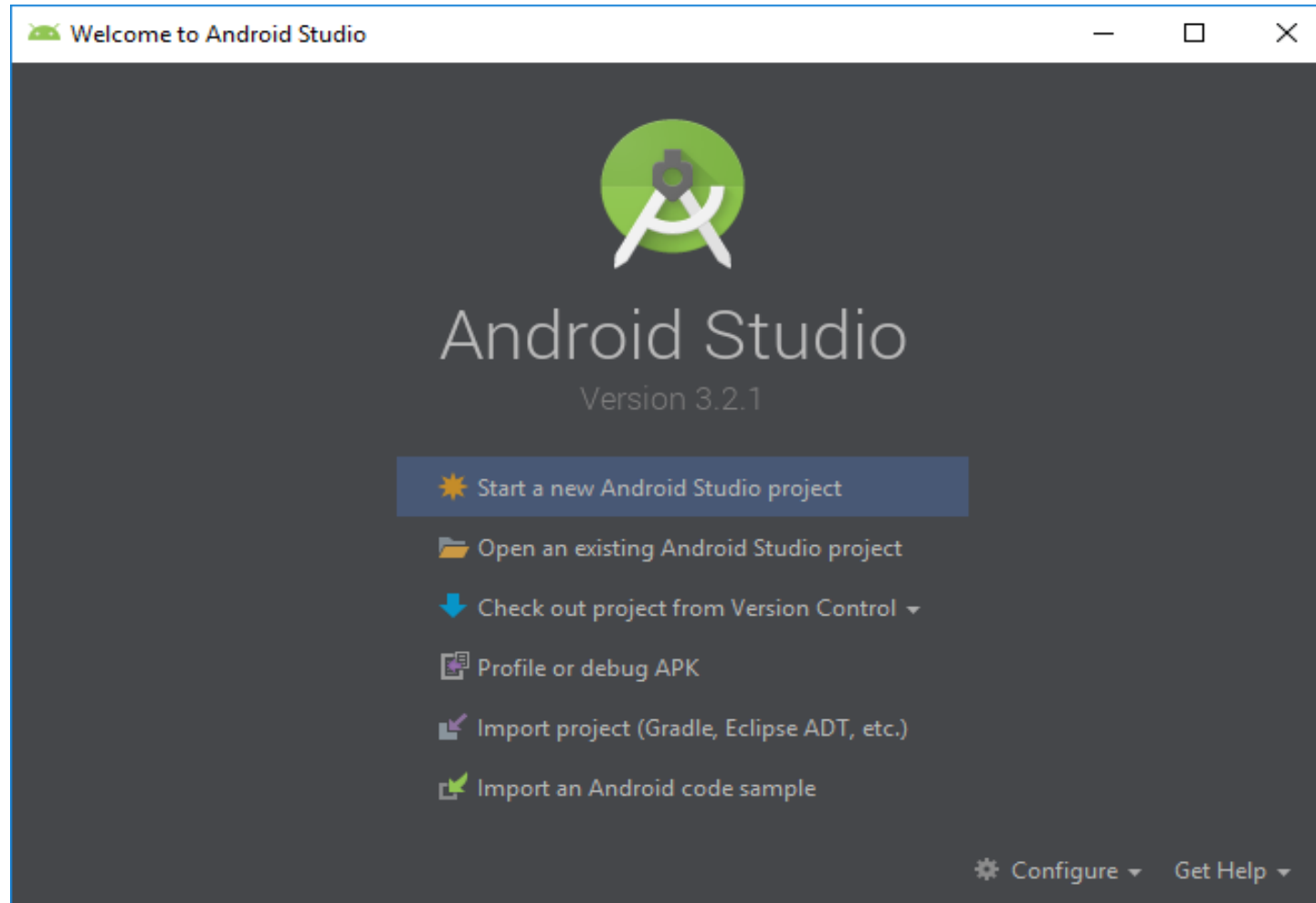
# Ferramentas de Desenvolvimento

Característica	Nativa	Web	Híbrida
Custo de desenvolvimento e manutenção	Alto	Baixo	Baixo
Tempo de desenvolvimento e manutenção	Alto	Baixo	Baixo
Desempenho	Alto	Baixo	Médio
Experiência do utilizador da aplicação móvel	Ótima	Baixa	Média
Reutilização de código em diferentes plataformas	Não	Sim	Sim
Acesso aos recursos do dispositivo	Total	Baixo	Parcial
Necessidade de conexão à internet	Não	Sim	Não
Canais de Distribuição	Lojas de aplicativo	Website	Lojas de aplicativo




**ANDROID STUDIO**

# Android Studio



# Android Studio

Create New Project

 Create Android Project

**Application name**

Boas Vindas

**Company domain**

cdm.mobile.edu

**Project location**

C:\dev\code\mobile\android\Boasvindas ...

**Package name**


edu.mobile.cdm.boasvindas Edit


☐ Include C++ support

☒ Include Kotlin support

Previous Next Cancel Finish

# Android Studio

 Create New Project

 Target Android Devices

### Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

☒ Phone and Tablet

API 15: Android 4.0.3 (IceCreamSandwich)

By targeting **API 15 and later**, your app will run on approximately **100%** of devices. [Help me choose](#)

☐ Include Android Instant App support

☐ Wear OS

API 23: Android 6.0 (Marshmallow)

☐ TV

API 21: Android 5.0 (Lollipop)

☐ Android Auto

☐ Android Things

API 24: Android 7.0 (Nougat)

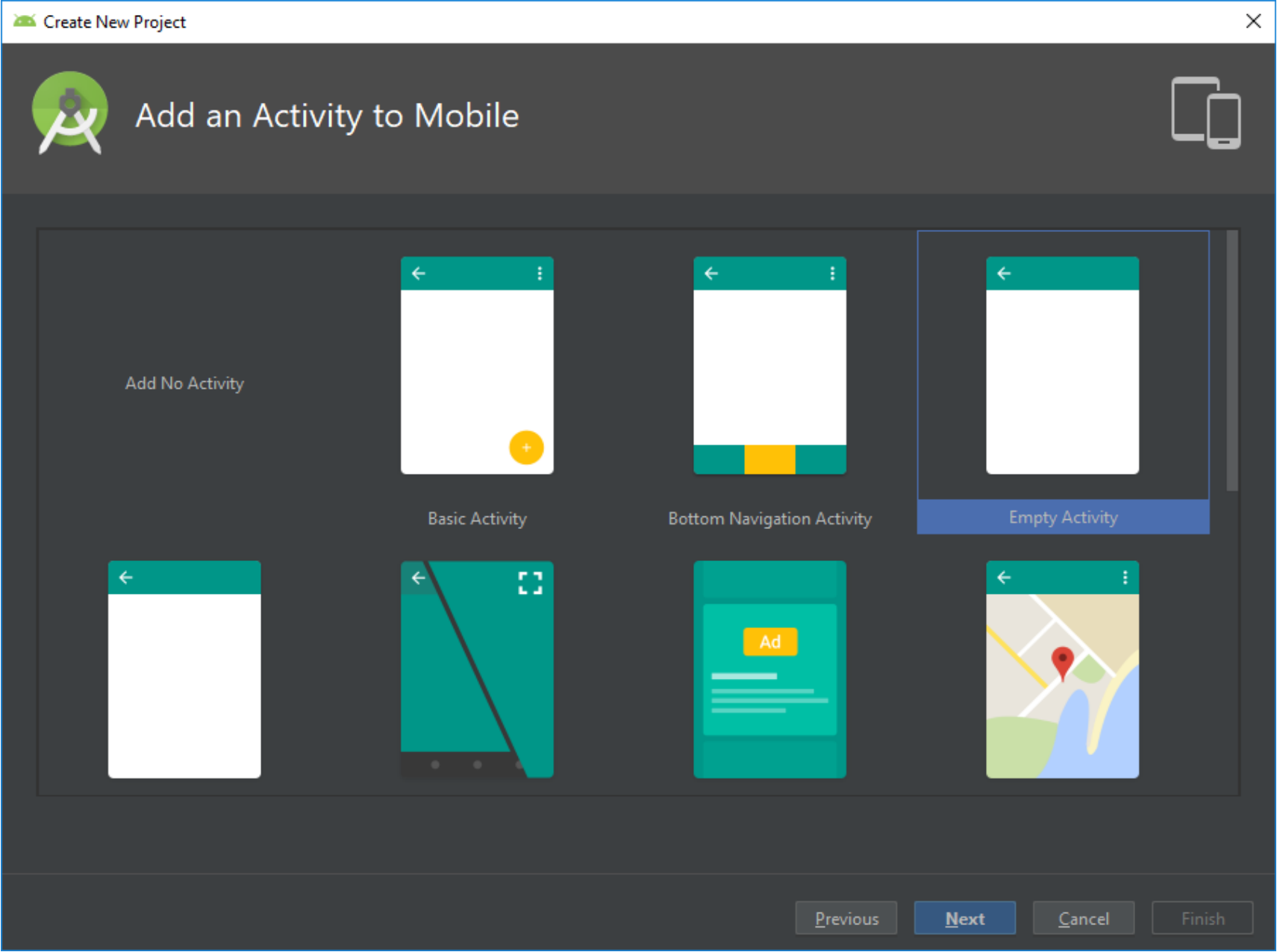
Previous

Next


Cancel


Finish


# Android Studio



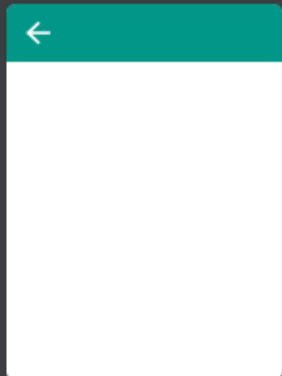
# Android Studio

 Create New Project

 Configure Activity



### Creates a new empty activity



Activity Name:

MainActivity

☒ Generate Layout File

Layout Name:

activity\_main

☒ Backwards Compatibility (AppCompat)

The name of the activity class to create

Previous

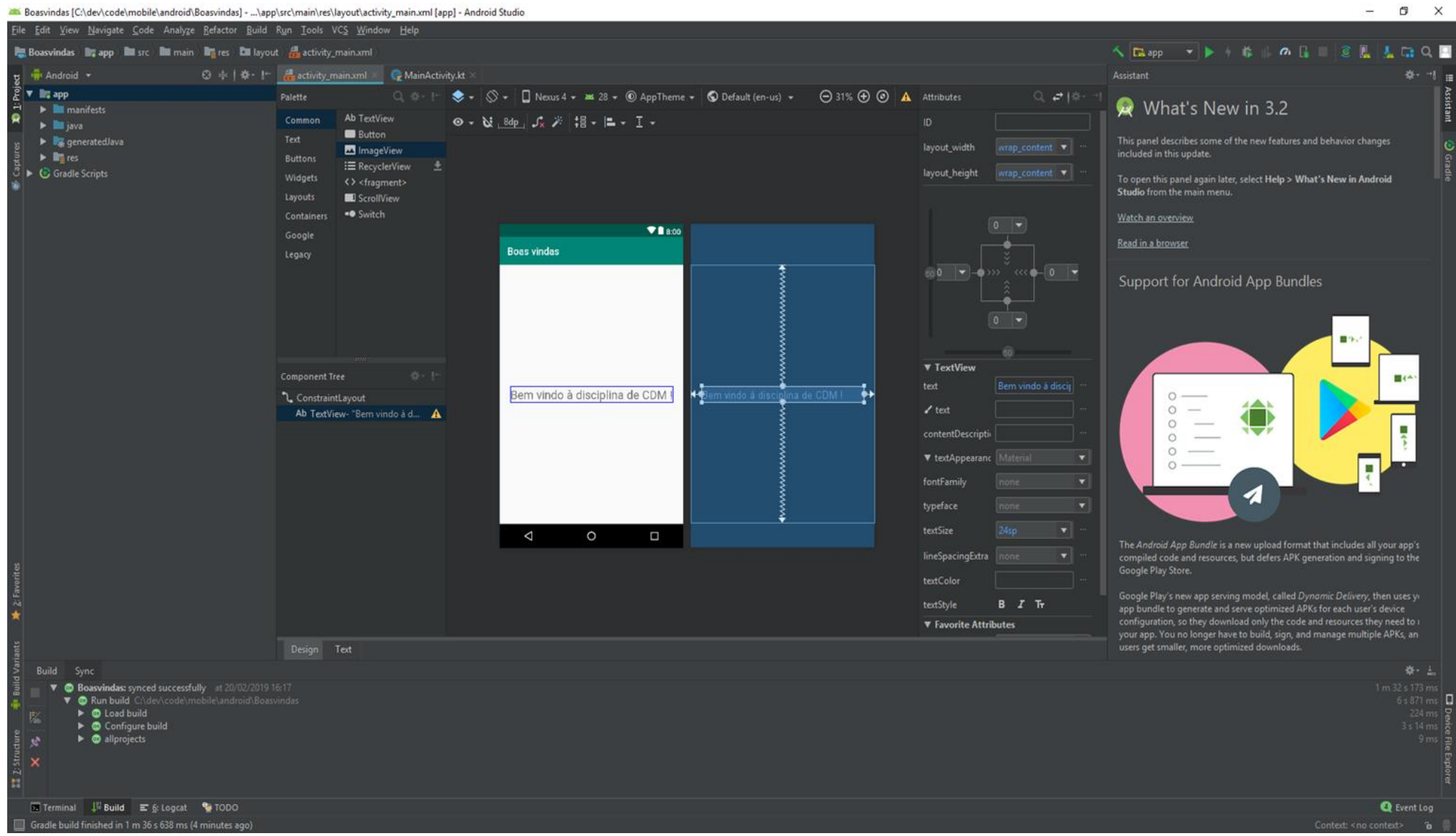
Next

Cancel

Finish



# Android Studio



# Aula 1

