

SERVICES ARCHITECTURE, API E MOBILE ARCHITECTURE

# ARQUITETURA MOBILE - *DESENVOLVIMENTO* HÍBRIDO E *NATIVO*

CHRISTIANE DE PAULA REIS



## LISTA DE FIGURAS

Figura 5.1 – The State of Mobile 2019 – Executive Summary.....	6
Figura 5.2 – Games Unique Position in Mobile .....	6
Figura 5.3 – Multiplataforma ou Nativo?.....	8
Figura 5.4 – Framework e plataformas.....	10
Figura 5.5 – Exemplo do componente Tabs do Ionic .....	12
Figura 5.6 – React Native Cross Platform .....	13
Figura 5.7 – Exemplo da estrutura do Cordova .....	15
Figura 5.8 – Ilustração de como o Cordova funciona para gerar os aplicativos .....	15
Figura 5.9 – Cordova App Architecture .....	16
Figura 5.10 – TypeScript .....	17

## LISTA DE QUADROS

Quadro 5.1 – Comparação de aplicativos híbridos e nativos .....	9
Quadro 5.2 – Top 5 <i>frameworks</i> mais utilizados .....	11

EMANIP

## SUMÁRIO

5 ARQUITETURA MOBILE – DESENVOLVIMENTO HÍBRIDO E NATIVO .....	5
5.1 Surgimento e evolução dos aplicativos mobile .....	5
5.2 Aplicações nativas .....	7
5.3 Aplicações híbridas .....	7
5.4 Nativo <i>versus</i> Híbrido .....	8
5.5 <i>Frameworks</i> para o desenvolvimento de aplicativos .....	10
5.6 Sobre o Ionic .....	11
5.6.1 Ionic components .....	12
5.6.2 Ionic Native .....	12
5.7 React Native .....	13
5.8 Entendendo sobre o Cordova .....	14
5.8.1 Arquitetura do Cordova .....	16
5.9 Um pouco sobre TypeScript .....	17
5.10 Bônus – Desenvolva pelo navegador! .....	18
5.11 CONCLUSÃO .....	18
REFERÊNCIAS .....	19
GLOSSÁRIO .....	20

## 5 ARQUITETURA MOBILE – DESENVOLVIMENTO HÍBRIDO E NATIVO

O crescente acesso à internet via dispositivos móveis e a popularização da banda larga trouxeram grandes problemas para o mundo da tecnologia, mas também foram grandes impulsionadores de mudança, levando ao surgimento de plataformas com maior flexibilidade e ferramentas que oferecem aos usuários controle sobre aplicações web.

Neste capítulo, você vai entender tudo o que envolve o mundo *mobile*.

### 5.1 Surgimento e evolução dos aplicativos mobile

Aplicativos, ou simplesmente apps, tornaram-se populares por meio dos smartphones. Desde 2007, a mania é desenvolver e disponibilizar apps que facilitem o dia a dia. Através de um app podemos pedir comida, acessar as câmeras de nossas casas, pedir um táxi, controlar uma empresa, comprar, vender, usar como entretenimento, entre muitas outras funcionalidades. Com tantas facilidades ao alcance das mãos, a tendência é que as pessoas passem cada vez mais tempo conectadas aos dispositivos móveis ou mobile.

Podemos definir aplicativos mobile como softwares que executam funcionalidades específicas para smartphones, tablets e desktops. E quando pensamos no custo e no prazo de desenvolvimento, foi constatado que é bem mais simples e rápido criar um aplicativo, além de envolver um menor gasto.

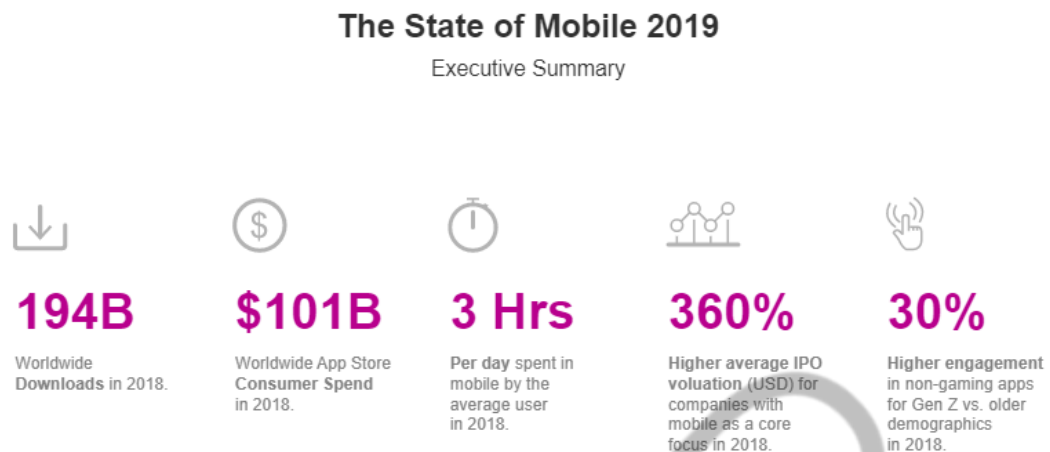


Figura 5.1 – *The State of Mobile 2019* – Executive Summary  
Fonte: App Annie (2019)

De acordo com o principal relatório de informações sobre dispositivos móveis do setor, *The State of Mobile 2019*, da App Annie – realizado em dezembro de 2018 –, o número de *downloads* globais de aplicativos no iOS e no Google Play foi de 194 bilhões em 2018, com um aumento de 35% de 2016 a 2018.

Os gastos do consumidor foram de US\$ 101 bilhões globalmente. Isso é impulsionado pelo crescente gasto feito pelo consumidor em jogos para celular – o formato de jogo mais popular e lucrativo – e também pela receita cada vez maior das assinaturas em aplicativos que não são de jogos. Os jogos representaram 33% dos *downloads* para dispositivos móveis e 74% dos gastos em 2018.

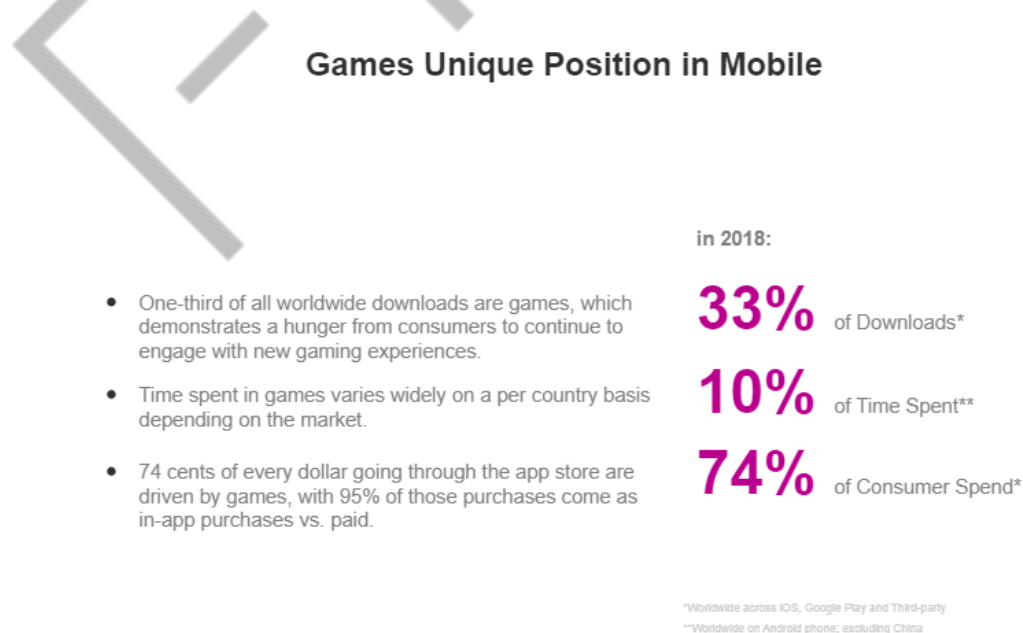


Figura 5.2 – Games Unique Position in Mobile  
Fonte: App Annie (2019)

Além disso, os dispositivos móveis ocuparam uma parcela maior do tempo dos consumidores em 2018; por exemplo, o usuário médio de smartphone nos Estados Unidos passou quase 3 horas por dia em aplicativos, um aumento de 10% em relação a 2017 e 20% em relação a 2016.

O celular não é mais um canal adicional – é o mecanismo que alimenta a transformação digital [...] O relatório *The State of Mobile 2019* libera o potencial do celular para empresas em todas as etapas da transformação do celular. As informações fornecidas nesse relatório são incomparáveis e informam a estratégia que toda empresa precisa para oferecer experiências vencedoras ao consumidor. (Theodore Krantz, CEO da App Annie)

Diante desse cenário, vamos mergulhar no mundo mobile para entender sobre o desenvolvimento mobile híbrido e nativo.

## 5.2 Aplicações nativas

**Aplicações nativas** são desenvolvidas para uma plataforma específica com sua respectiva linguagem. Por exemplo, para a plataforma iOS, você escreve código em Swift, e para a plataforma Android, você escreve código em Kotlin ou Java, dessa forma, seu aplicativo é denominado como um aplicativo nativo.

## 5.3 Aplicações híbridas

**Aplicações híbridas** são apps web que usam tecnologias da web como HTML5, CSS e JavaScript a fim de desenvolver um aplicativo para diferentes plataformas, como Android, iOS, Windows (*Universal Windows Platform* – UWP) ou qualquer outra plataforma.

O conceito híbrido é dado pelo fato de que sua interface não apresenta componentes nativos. Na verdade, o aplicativo híbrido é uma aplicação que possui apenas uma WebView em toda a tela, para gerar seu HTML e CSS, com JavaScript manipulando esses componentes e apresentando-se como uma aplicação responsiva.

Para fornecer uma experiência nativa ao usuário e acessar recursos nativos das diferentes plataformas, é necessário fazer uso de *frameworks* ou estruturas que

acessam diretamente os componentes nativos. Por exemplo, ao utilizar o *framework* PhoneGap, que é uma distribuição do Cordova, você tem acesso a componentes nativos como GPS, entre tantos outros.

Aplicações híbridas são aplicações multiplataforma, ou seja, você desenvolve apenas um código para disponibilizar seu produto para iOS e Android simultaneamente, sem a necessidade de desenvolver duas vezes, usando a linguagem nativa de cada plataforma.

**IMPORTANTE:** Multiplataforma (ou *cross-platform*) significa que um único código-fonte do aplicativo é compilado em código nativo de diferentes sistemas operacionais, como Android, iOS, UWP, entre outros.

#### 5.4 Nativo *versus* Híbrido



Figura 5.3 – Multiplataforma ou Nativo?  
Fonte: Google Imagens (2019)

Sempre há aquela famosa dúvida: para quem devo desenvolver, híbrido ou nativo? E, como tudo na vida, a resposta é: depende. Aplicativos híbridos não são balas de prata, eles possuem vantagens e desvantagens. O quadro a seguir ilustra as vantagens e as desvantagens do desenvolvimento de aplicativos híbridos e nativos.



Requisito	Nativo	Híbrido
<b>Controle no dispositivo mobile</b>	Possui controle total dos recursos que o dispositivo oferece, como acelerômetro, giroscópio, geolocalização etc.	Possui controle da maioria dos componentes nativos, porém ainda não possui controle total dos recursos.
<b>É multiplataforma?</b>	Não. Requer um desenvolvimento específico para cada plataforma.	Sim. Você desenvolve uma única vez para qualquer tipo de plataforma.
<b>Requer IDE específica para desenvolver apps?</b>	Sim. Como exemplo, apps para iOS utilizam a IDE Xcode e linguagem Swift; e apps para Android utilizam a IDE Android Studio e linguagem Java ou Kotlin.	Não. É possível desenvolver com o bloco de notas, mas existem diversos editores que facilitam o trabalho, como o Atom, Visual Studio Code, Sublime Text, VIM, entre muitos outros.
<b>Baixo custo e rápida produção?</b>	Não. Deve ser desenvolvido em diferentes linguagens, sendo necessário contratar um desenvolvedor específico para cada plataforma. Seu custo é elevado.	Sim. Não necessita de equipes de desenvolvimento separadas para cada plataforma. E por possuir uma única produção e manutenção, ou seja, um único código é capaz de gerar o app para ambas as plataformas, seu custo se torna mais reduzido.
<b>Interface do usuário (UI) e Experiência do usuário (UX)</b>	Explora muito bem toda a UI/UX, seguindo o padrão de interface do usuário adequadamente.	Com a evolução da tecnologia, as funcionalidades de interface web chegam ao mesmo nível que componentes de interface nativos, porém, ainda assim, continuam apenas sendo uma WebView.
<b>Performance</b>	Ótimo desempenho.	Depende muito dos requisitos que sua aplicação vai possuir.

Quadro 5.1 – Comparação de aplicativos híbridos e nativos  
 Fonte: Elaborado pela autora (2019)

Como o quadro mostra, não há um desenvolvimento melhor que o outro, isso vai depender muito do projeto e/ou da demanda.

No nosso cotidiano, temos bons exemplos de grandes empresas e startups multinacionais que alcançaram sucesso por meio de aplicativos que foram desenvolvidos nativamente, como o Facebook Messenger, o WhatsApp e o Skype.

Porém, podemos concluir que, para projetos simples, sem muito acesso nativo e com tempo curto de desenvolvimento, com toda a certeza, aplicativos híbridos são a melhor escolha.

### 5.5 *Frameworks* para o desenvolvimento de aplicativos

Os *frameworks* são os grandes responsáveis pela expansão do desenvolvimento de apps multiplataforma, os conhecidos aplicativos híbridos, pois é por meio do *framework* que acontece a conversão de um app Android em um app para iOS e vice-versa.

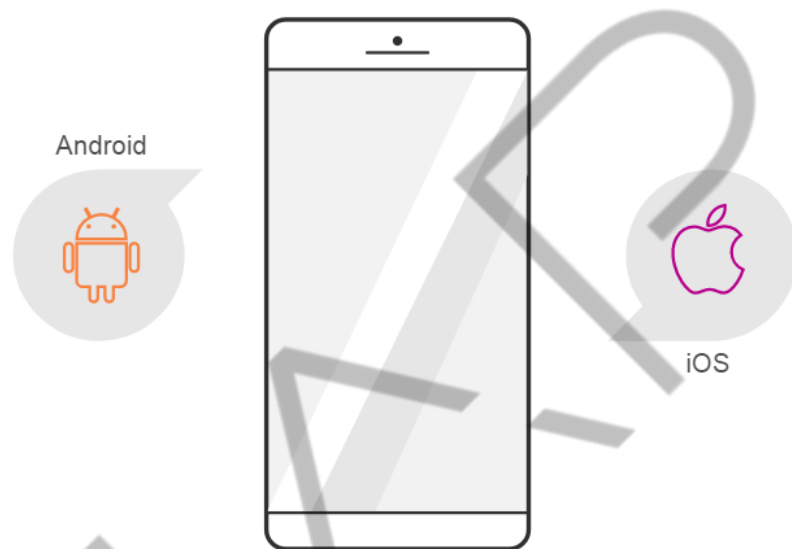


Figura 5.4 – *Framework* e plataformas  
Fonte: Elaborado por FIAP (2019)

Os top 5 *frameworks* mais utilizados são:

<i>Framework</i>	<i>Tipo de aplicativo gerado</i>	<i>Funcionalidades</i>
<b>ionic</b>	Nativo / Híbrido	Cria aplicativos nativos com uma combinação de HTML, CSS3 e AngularJS ou JavaScript; e aplicativos híbridos com HTML5. É considerado um dos melhores <i>frameworks</i> para o desenvolvimento dos <i>Progressive Web Apps</i> (PWA). É <i>open-source</i> e gratuito.
<b>React Native</b>	Nativo	Um dos melhores <i>frameworks</i> JavaScript para criar aplicativos nativos para diversas plataformas. Suporta Node.js. Permite escolher entre o React, para web, e o React Native para o desenvolvimento de aplicativos móveis. Foi lançado pelo Facebook.
<b>Xamarin</b>	Híbrido	Cria aplicativos nativos com base de código em C#. Foi lançado pela Microsoft. É gratuito.
<b>PhoneGap</b>	Híbrido	Cria aplicativos híbridos com uma combinação de HTML5, CSS3 e JavaScript. Oferece o melhor desempenho da categoria e não possui restrição de hardware. Permite adição de plugins nativos. A estrutura do Adobe PhoneGap é uma distribuição <i>open-source</i> do Apache Cordova.
<b>Flutter</b>	Híbrido	Cria aplicativos híbridos para as plataformas Android e iOS usando linguagem Dart. Faz uso do mecanismo de renderização 2D, chamado Skia, fornecendo desempenho nativo completo no iOS e no Android. Foi lançado pela Google. É <i>open-source</i> e gratuito.

Quadro 5.2 – Top 5 *frameworks* mais utilizados  
 Fonte: Elaborado pela autora (2019)

Na hora de escolher o *framework* a ser utilizado, considere aquele que melhor se adequar às necessidades do seu projeto ou aplicativo, visto que cada *framework* possui características próprias.

## 5.6 Sobre o Ionic

O Ionic é um *framework* totalmente baseado no Angular e focado em gerar componentes com interface gráfica voltada para usabilidade mobile. Possibilita integração com outras bibliotecas ou estruturas, como o Angular, React e Vue, ganhando um grande destaque em seu estilo para cada dispositivo. Porém usa o WebView a fim de oferecer uma experiência nativa.

O Ionic também se destaca pela abstração total com a comunicação do Cordova. Com isso, temos também a API do Ionic, que trabalha exatamente com os plugins Cordova, chamados Ionic Native.

### 5.6.1 Ionic components

O Ionic tem a grande vantagem de possuir componentes com design e serviços adaptados já para as principais plataformas, como Android, iOS, UWP e PWA.

Com sua nova versão junto à do Angular, todos os componentes de design foram refeitos e aperfeiçoados para cada sistema operacional, tornando-o, assim, quase imperceptível para aplicações nativas.

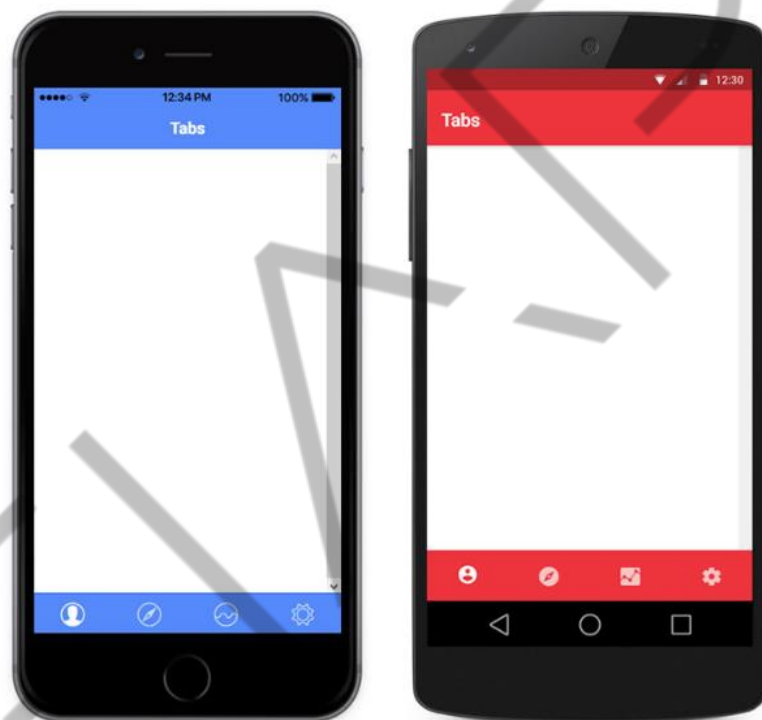


Figura 5.5 – Exemplo do componente Tabs do Ionic  
Fonte: Google Imagens (2019)

Saiba mais sobre os componentes:

<<https://ionicframework.com/docs/components/>>.

### 5.6.2 Ionic Native

Ionic Native é uma biblioteca gratuita de plugins e integrações do Cordova adaptados para TypeScript que facilitam o desenvolvimento de funcionalidades nativas para qualquer aplicativo Ionic, projeto Cordova ou WebView. São mais de 200 plugins nativos que você pode inserir facilmente no seu aplicativo, com poucas linhas de códigos.

Dúvida? Aqui está um pequeno exemplo de geolocalização: <<https://fiapon-ionic-geolocation.stackblitz.io>>. Alguns serviços já conseguem utilizar a mesma tecnologia tanto nativo quanto para web, esse exemplo funciona tanto em aplicativos nativos quanto em navegadores.

Saiba mais sobre a biblioteca em: <<https://ionicframework.com/docs/native/>>.

## 5.7 React Native

O **React Native** é um *framework cross platform* (ou multiplataforma) que está em alta com o time dos *devs*, pois permite criar aplicativos móveis interativos para as plataformas iOS, Android e UWP, com uma única base de código escrita em JavaScript, e não exige equipes de desenvolvimento separadas para cada plataforma. Essa tecnologia não trabalha com HTML ou CSS, os estilos seguem o *layout* padrão com Flexbox e são escritos em linha.

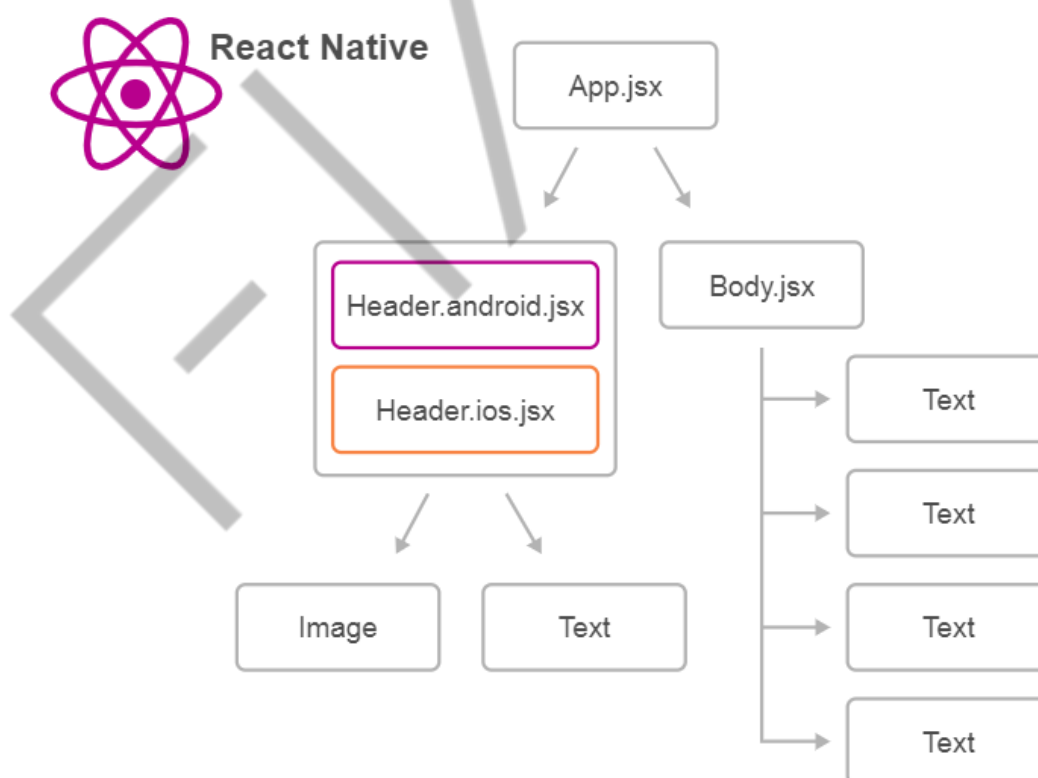


Figura 5.6 – React Native Cross Platform  
Fonte: Google Imagens (2019)

React Native se baseia em React e possui seu próprio empacotador. Todos os arquivos JavaScript são agrupados em um pacote único que pode ser consumido

e executado por diferentes mecanismos. Como, por exemplo, o JavaScriptCore, que é o mecanismo JavaScript da Apple usado tanto para iOS quanto para Android; e o ChakraCore, que é o mecanismo JavaScript da Microsoft usado para aplicativos UWP. O React é uma biblioteca JavaScript declarativa que possui uma estrutura mais estável e é ideal para aplicações em larga escala. Fornece excelente desempenho e capacidade de resposta.

Assim como o Ionic, o React Native também pode se integrar ao Cordova por meio de plugin específico. Como diferencial, além de possibilitar reúso de código, diminuindo consideravelmente o tempo de desenvolvimento, o aplicativo híbrido é renderizado nativamente por causa das APIs da plataforma (os componentes reutilizáveis) e o resultado é uma interface de usuário nativa de alta qualidade.

Algumas empresas que desenvolveram seus aplicativos com o React Native são Facebook, Uber Eats, Yahoo!, Instagram, AirBnB, Nubank, entre outras.

## 5.8 Entendendo sobre o Cordova

O Cordova é uma estrutura de desenvolvimento *open-source* com base para a criação de aplicativos híbridos para mobile, que atua como *backend* no seu aplicativo. Permite usar tecnologias padrão da Web, como HTML5, CSS3 e JavaScript para criar aplicações mobile *cross platform* para diversas plataformas (Android, iOS, UWP), além de comunicar-se por meio de componentes nativos, como, por exemplo, câmera, GPS, acelerômetro, geolocalização etc. Porém sua interface gráfica é gerada por um WebView.

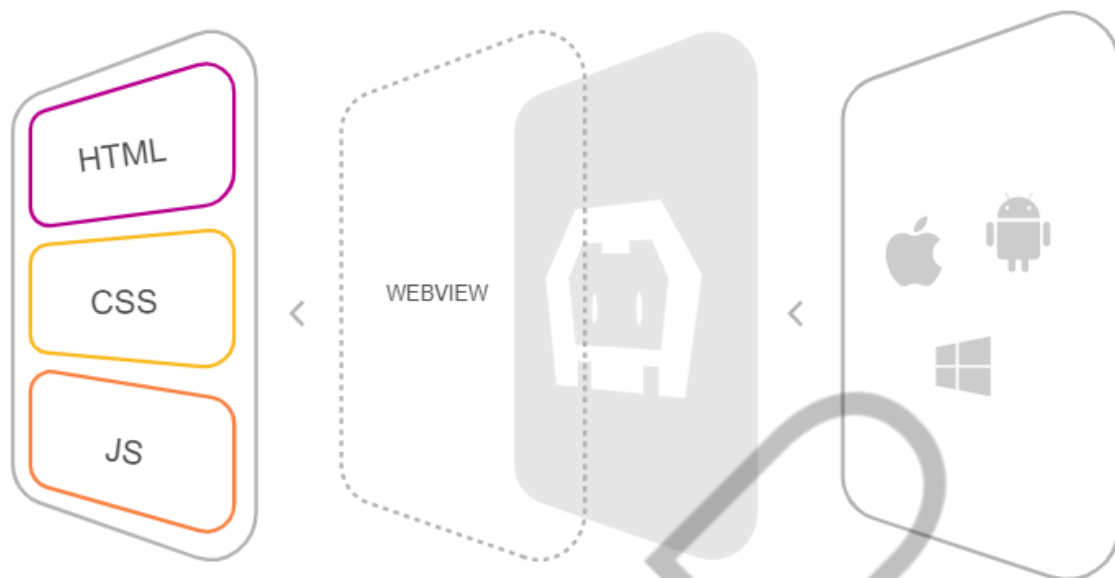


Figura 5.7 – Exemplo da estrutura do Cordova  
Fonte: Google Imagens (2019)

O Cordova é o *core* de toda aplicação híbrida e você pode desenvolver apenas com HTML5, CSS e JavaScript Vanilla. É o Cordova que permite que seu aplicativo seja transformado em “pacote nativo” para cada sistema operacional.

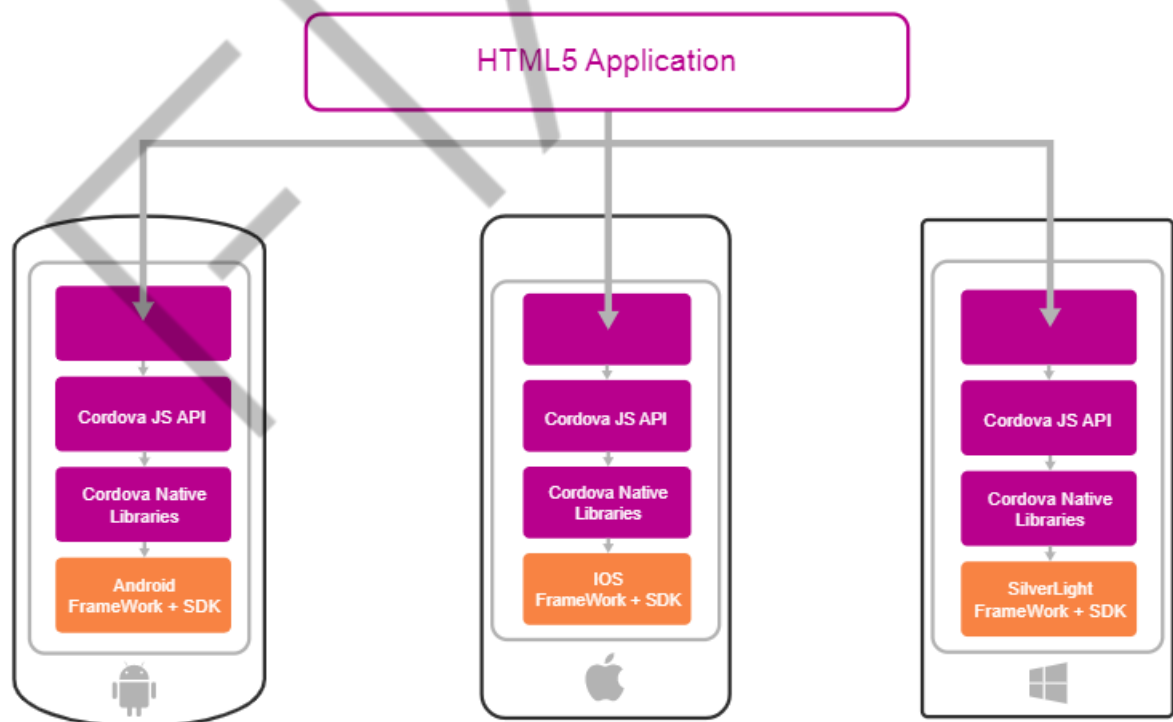


Figura 5.8 – Ilustração de como o Cordova funciona para gerar os aplicativos  
Fonte: FIAP (2019)

Veja, a seguir, como é a arquitetura do Cordova e seus componentes.

### 5.8.1 Arquitetura do Cordova

A arquitetura do Cordova é composta basicamente por três *containers* de componentes, necessários para compor um aplicativo, que são o **WebView**, o **Web App** e os **Plugins**. O diagrama a seguir mostra uma visualização de alto nível da arquitetura do aplicativo Cordova.

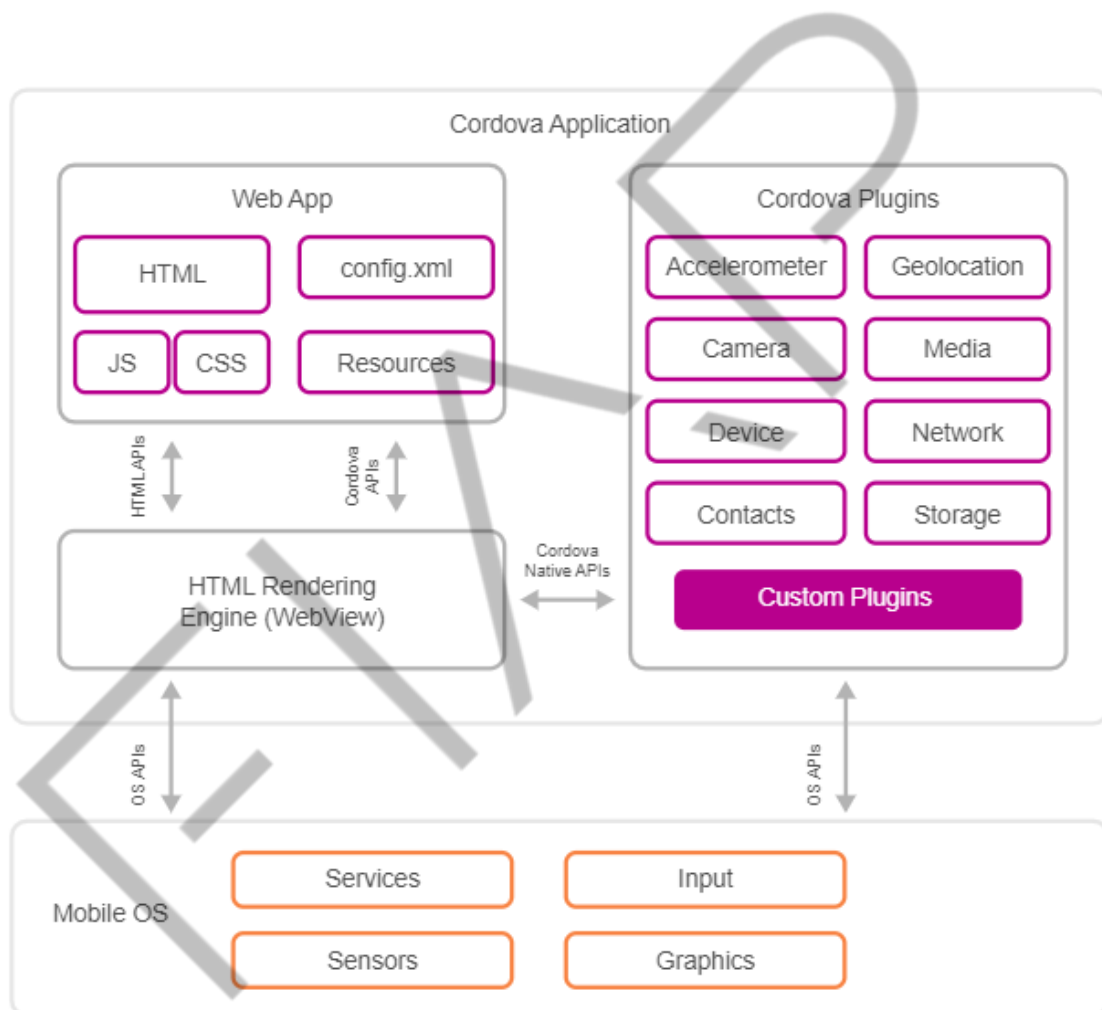


Figura 5.9 – Cordova App Architecture  
Fonte: Apache Cordova (2019)

**IMPORTANTE:** Quando você cria um projeto Cordova, ele não possui plugins. Quaisquer plugins que você deseje, mesmo os principais, devem ser adicionados explicitamente.

**WebView:** pode fornecer ao aplicativo toda a interface do usuário ou pode ser um componente em um aplicativo híbrido maior que combina o WebView com



componentes de aplicativos nativos. Vai depender da plataforma e da necessidade do seu projeto.

**Web App:** é um container que abriga todo o código do aplicativo. Dentro dele, existe o arquivo `config.xml`, responsável por fornecer informações cruciais sobre o aplicativo e por especificar parâmetros a respeito do seu funcionamento.

O aplicativo em si é implementado como uma página da Web que faz referência a CSS, JavaScript, imagens, arquivos de mídia ou outros recursos necessários para sua execução. Por padrão, esse arquivo é chamado de `index.html`. O aplicativo é executado em um `WebView` no *wrapper* do aplicativo nativo, que você distribui para as lojas de aplicativos.

**Plugins:** são parte integrante do ecossistema Cordova. Eles fornecem uma interface para o Cordova e os componentes nativos se comunicarem entre si; e para as ligações se comunicarem com as APIs de dispositivo padrão. Isso permite que você chame o código nativo do JavaScript.

## 5.9 Um pouco sobre TypeScript

TypeScript é um *superset open-source* de JavaScript desenvolvido pela Microsoft. Além de adicionar novas funcionalidades e melhorar algumas já existentes no ES6 (EcmaScript6 ou ES2015), o TypeScript é responsável por compilar um código JavaScript. Dessa forma, a linguagem pode relatar erros de compilação, facilitando muito o processo de tratamento de erros.

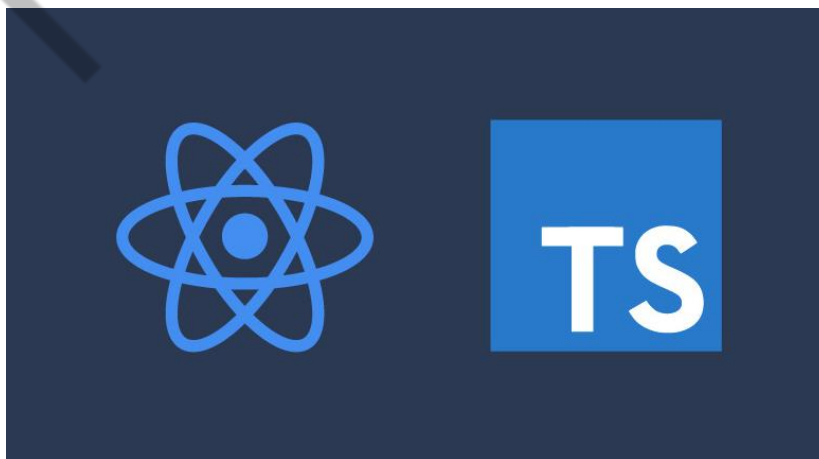


Figura 5.10 – TypeScript  
Fonte: Google Imagens (2019)

### 5.10 Bônus – Desenvolva pelo navegador!

Uma das maiores vantagens para o desenvolvimento de frameworks web é que existem muitas plataformas on-line de desenvolvimento. Uma das minhas favoritas é o [StackBlitz](#), que é totalmente baseada no Visual Studio Code e suporta desenvolvimento de aplicações Angular, Ionic e React.

### 5.11 CONCLUSÃO

Até o momento, você adquiriu todo o conhecimento sobre o que são aplicações híbridas e nativas, um pouco do que é Cordova e, por fim, aprendeu sobre o Ionic e React Native. Agora você está preparado para desenvolver seu primeiro aplicativo mobile. Então, vamos botar a mão no código!

## REFERÊNCIAS

APACHE CORDOVA. **Overview.** Disponível em: <<https://cordova.apache.org/docs/en/latest/guide/overview/index.html>>. Acesso em: 10 out. 2019.

APP ANNIE. **The State of Mobile in 2019 report.** 2019. Disponível em: <<https://www.appannie.com/en/about/press/releases/app-annie-releases-annual-state-of-mobile-2019-report/>>. Acesso em: 10 out. 2019.

IONIC ACADEMY. **Ionic QR Code Generator & Reader [v3].** 2017. Disponível em: <<https://ionicacademy.com/ionic-qr-code-generator-reader/>>. Acesso em: 10 out. 2019.

IONIC FRAMEWORK. **Ionic 4 is coming.** [s.d.]. Disponível em: <<https://ionicframework.com/docs/>>. Acesso em: 10 out. 2019.

REACT NATIVE FRAMEWORK. **React Native0.57 – Getting Started.** Disponível em: <<https://react-native.org/doc/getting-started.html>>. Acesso em: 10 out. 2019.

## GLOSSÁRIO

<b>PWA – Progressive Web Apps</b>	É uma metodologia de desenvolvimento de software que utiliza os melhores recursos da Web para entregar ao usuário uma experiência tão boa quanto a de um aplicativo nativo.
<b>Flexbox</b>	É um modelo de layout da Web CSS3. Foi projetado para fornecer um <i>layout</i> consistente em diferentes tamanhos de tela, permitindo funcionar da mesma maneira tanto no React Native quanto no CSS na Web.
<b>WebView</b>	É um componente nativo de aplicações Android e iOS, que consegue gerar conteúdos HTML dentro de aplicativos.
<b>Vanilla</b>	É a linguagem pura, sem possuir nenhum tipo de biblioteca ou <i>framework</i> .
<b>UWP – Universal Windows Platform</b>	É a plataforma da Microsoft que permite que o mesmo aplicativo seja executado no Windows Phone 10 Mobile, Xbox One e Windows 10.