

## VANTAGENS E DESVANTAGENS DA UTILIZAÇÃO DO *IONIC* FRAMEWORK PARA O DESENVOLVIMENTO DE APLICATIVOS MÓVEIS

Neri Norberto Lange Jr.\*

Neyza Bibiana Guzman Mercado\*\*

**Resumo.** Este artigo apresenta as vantagens e desvantagens da utilização do *Ionic Framework* para desenvolvimento de aplicativos móveis. O artigo é escrito de modo explicativo e objetivo, proporcionando ao leitor, informações sobre o mercado de dispositivos móveis, bem como os principais conceitos relacionados à ferramenta *Ionic* e todas as tecnologias utilizadas por ela, possibilitando ao leitor conhecer esta tecnologia e através de suas vantagens e desvantagens saber quando utilizá-la.

Palavras-chave: *Ionic*. Aplicativos móveis. Ferramentas. Tecnologias. Vantagens. Desvantagens.

**Abstract.** This article presents the advantages and disadvantages of the use of the *Ionic Framework* for mobile application development. The article is written so explanatory and goal, providing the reader, information about the market of mobile devices, as well as the main concepts related to *Ionic* tool and all the technologies used by her, enabling the reader to know this technology and its advantages and disadvantages learn when and how to use it.

Keywords: *Ionic*. Mobile applications. Tools. Technologies. Advantages. Disadvantages.

---

\* Estudante – Cursando Ciência da Computação – Unifavest – junior.masterinfo@gmail.com

\*\* Professora Orientadora – Bacharel em Ciência da Computação – UFSCar  
Msc. em Engenharia de Produção e Sistemas – UFSC – neyzabibiana@hotmail.com

## 1 INTRODUÇÃO

Em um mundo cada vez mais conectado, é evidente o crescimento constante do uso de *smartphones*. Em casa, no ônibus, nas salas de espera, no trabalho, até mesmo em lugares onde a atenção deveria estar focada em outras áreas, como por exemplo, no trânsito. O fato é que as pessoas parecem não viver mais sem seus *smartphones*.

Dentre os aplicativos mais utilizados estão às redes sociais, os aplicativos de troca de mensagens, bancos, e-mail, compras, mapas e localização. Esses dados podem parecer um pouco vagos de início, mas quando se trata de elaborar estratégias para uma marca ou empresa, eles se mostram muito úteis. O uso das redes sociais, por exemplo, mostra onde as pessoas gastam mais tempo e dedicam grande parte da sua atenção, conseqüentemente são espaços que não devem ser nunca ignorados pelos anunciantes.

Visando suprir as necessidades dos clientes, as empresas estão cada vez mais investindo em soluções que estejam rapidamente acessíveis para os consumidores. Em 2022, espera-se que 70% de todas as interações de *software* corporativo seja através de dispositivos móveis. (IONIC, 2018)

O desenvolvimento dessas aplicações, não é tarefa fácil. O mercado mobile está dividido entre três sistemas operacionais: Android<sup>1</sup>, IOS<sup>2</sup> e Windows Phone<sup>3</sup>, onde cada um possui características e arquiteturas distintas, e o desenvolvimento amplamente difundido para utilizar a linguagem nativa e desenvolver uma aplicação específica para cada sistema operacional (SILVA, 2015).

Isso acaba aumentando muito o tempo e o custo para desenvolver tal aplicação. Devido a esta grande demanda muitas equipes de desenvolvedores de aplicativos nativos estão com dificuldades para acompanhar este mercado.

---

<sup>1</sup> Sistema operacional de celular desenvolvido e mantido pela empresa Google.

<sup>2</sup> Sistema operacional de celular desenvolvido e mantido pela empresa Apple.

<sup>3</sup> Sistema operacional de celular desenvolvido e mantido pela empresa Microsoft.

Para suprir esta necessidade, os profissionais da área de desenvolvimento devem estar cada vez mais preparados para a produção de soluções funcionais, intuitivas ao usuário e com baixo consumo de recursos físicos (*hardware*). Estes pontos devem estar alinhados ao baixo custo de produção e tempo gasto durante o desenvolvimento.

A produtividade é um dos pontos mais valorizados no processo de desenvolvimento de aplicativos móveis, por este motivo, os desenvolvimentos de aplicativos híbridos podem ser uma boa alternativa para o mercado atual. Este tipo de tecnologia reduz os custos, pois através de único código-fonte, pode-se exportar o aplicativo para diversas plataformas, necessitando-se de menos tempo de desenvolvimento, e a empresa não necessita de uma equipe de desenvolvimento experiente em cada plataforma.

O *Ionic framework* é uma plataforma em crescimento no mercado pela sua alta produtividade e a facilidade na manipulação de seus recursos, visto que muitos profissionais aprendem ainda na faculdade os conceitos utilizados pela plataforma. Através da combinação de *JavaScript*, *HTML*, *CSS*, *Angular*, *TypeScript*, *JavaScript*, *Node* e *Cordova*, um mesmo aplicativo pode ser disponibilizado nas plataformas *Android*, *iOS* e *Windows Phone* com simples comandos disponibilizados pelo *framework*.

## 2 REFERENCIAL TEÓRICO

### 2.1 Aplicativos Nativos x Aplicativos Híbridos x *Progressive Web App*

Ao se desenvolver um determinado aplicativo, algumas perguntas devem ser respondidas. Entre elas: Qual a necessidade do cliente? Quais funcionalidades serão utilizadas? Quanto tempo de desenvolvimento estimado? Qual o orçamento para este projeto?

Com base nestas respostas é possível definir que tipo de aplicativo será desenvolvido, nativo, híbrido ou com uso de *progressive app*.

O aplicativo nativo é desenvolvido em linguagem exclusiva de cada sistema operacional, como Java<sup>4</sup> no Android, Objective-C<sup>5</sup> no iOS e C#<sup>6</sup> no Windows Phone. Cada plataforma apresenta suas próprias ferramentas e elementos de interface.

Por ser programado exclusivamente para um determinado sistema operacional, apresenta e disponibiliza mais recursos ao usuário, por conseguir utilizar todos os recursos oferecidos pelo smartphone, como câmera, GPS, leitor de digital entre outros. Também costuma seguir um padrão de design oferecido por cada sistema operacional, os chamados guias de design, que contém melhores práticas para proporcionar uma melhor experiência ao usuário.

Essa programação exclusiva faz com que o desempenho do aplicativo nativo seja superior e seu tempo de utilização seja maior que os demais.

Alguns exemplos de aplicativos nativos são: WhatsApp, Facebook, Instagram e Uber.

O custo do aplicativo nativo pode ser mais elevado, pois basicamente tudo é desenvolvido em duas fases, por mais que o *back-end* possa ser o mesmo, devem ser desenvolvidas várias versões de *front-end*, uma para cada sistema operacional.

Para publicar o aplicativo nas lojas é necessário uma aprovação, ou seja, ele precisa estar de acordo com as políticas internas de cada uma delas. Além disso, normalmente, o *app* nativo também funciona no modo *offline*. (BATTISTELLI, 2018)

O aplicativo híbrido de maneira bem básica é uma página web embutida em um aplicativo nativo. É desenvolvido em *HTML*, *CSS* e *JavaScript*, onde através de ferramentas pode ser convertido para IOS, Windows Phone e Android. Isso significa que qualquer pessoa com um conjunto de habilidades do desenvolvedor da *web* pode criar um aplicativo usando a abordagem híbrida.

As ferramentas como o *Cordova Framework* e o *Ionic Framework* disponibilizam uma série de *plug-ins* que fazem a “tradução” dos códigos. Trata-se de

---

<sup>4</sup> Linguagem de programação utilizada para desenvolvimento de aplicativos Android.

<sup>5</sup> Linguagem de programação utilizada para desenvolvimento de aplicativos IOS – Apple.

<sup>6</sup> Linguagem de programação utilizada para desenvolvimento de aplicativos Windows Phone.

uma linguagem padrão para ambas as plataformas, onde cada produto final é apresentado no formato visual do sistema operacional nativo.

A produtividade é umas das características mais impactantes quando se trata de aplicativos híbridos. Por conseguir a partir de apenas um código-fonte gerar a portabilidade da aplicação para as principais plataformas, os custos com tempo de desenvolvimento e recursos humanos necessários é reduzido consideravelmente, visto que não são necessários profissionais dedicados para cada plataforma disponível no mercado.

Segundo Battistelli (2018) um exemplo de aplicativo híbrido muito bem construído é o do Netflix, que utiliza a mesma interface para todos os dispositivos.

Segue abaixo um quadro que faz algumas comparações importantes sobre ambas as tecnologias.

Quadro 1 – Comparação entre aplicativos híbridos e nativos

<b>Características</b>	<b>Nativo</b>	<b>Híbrido</b>
<b>Linguagem de programação</b>	Objective C, Java, .net	HTML5, CSS, JavaScript
<b>Armazenamento off-line</b>	Possui	Possui
<b>Notificações</b>	Possui	Possui
<b>Distribuição</b>	App Store	App Store
<b>Geolocalização</b>	Possui	Possui
<b>UI/UX</b>	Melhor	Mediana
<b>Custo</b>	Mais caro	Mais barato

Fonte: Do Autor (2018)

Em meados de 2016, começou-se a estudar uma nova tecnologia para desenvolvimento de aplicativos, chamada de *Progressive Web App* (aplicativo progressivo para a *web*), mas que ainda não se popularizou totalmente no mercado.

Trata-se de um *website* na versão *mobile*, utilizado em qualquer navegador, como por exemplo o Chrome e à medida em que você interage com a página, você pode adicionar o *web app* à seu menu de aplicativos no celular. Após adicionado, ele

passa a adquirir funções que antes eram exclusivas de aplicativos nativos como geolocalização, notificações, uso off-line, além de deixar um ícone no menu de seu celular, como os aplicativos híbridos e nativos fazem.

Independentemente do navegador utilizado, o usuário tem todas as funcionalidades do aplicativo disponíveis. Assim, a interface é a mesma de um aplicativo, mas a velocidade é igual a de um site. (RANK MY APP, 2018)

Neste tipo de aplicação, também não é necessário fazer o *download* do aplicativo em lojas como na Apple Store ou no Google Play, sendo assim o usuário pode primeiro utilizar a tecnologia, para posteriormente adicioná-la a seu dispositivo, processo no qual ocorre de maneira inversa nos aplicativos nativos e híbridos, onde primeiro o usuário precisa instalar o *app*, para posteriormente conseguir testá-lo.

Segundo Russell (2015) este tipo de aplicação além de propiciar vantagens aos usuários, também contêm uma série de vantagens para os desenvolvedores, como:

- Responsivo: se encaixa mais facilmente em qualquer resolução de tela.
- Independente de conexão: o aplicativo pode funcionar até quando o usuário está *offline*.
- Interações avançadas como de *apps*.
- Sempre atualizado: o usuário não precisa “baixar uma atualização do *app*”.
- Seguro: o conteúdo do *app* é servido com tecnologias para prevenir intrusos.
- *SEO-friendly*: os mecanismos de busca conseguem encontrar o conteúdo dos aplicativos.
- Instalável: podem ser adicionados à *home screen* do celular, permitindo que os usuários “salvem” os aplicativos que eles considerarem mais úteis ou importantes.
- Linkável: mais fáceis de compartilhar conteúdo ao enviar o link para alguém.

Este novo tipo de desenvolvimento ainda é pouco conhecido no mercado brasileiro, mas grandes empresas como Facebook e Google estão interessadas no

projeto, fazendo com que conseqüentemente nos próximos anos, hajam novidades nesta área.

## 2.2 Ionic Framework

Raramente um aplicativo para dispositivos móveis é projetado apenas para uma única plataforma. Consumidores, parceiros, e todos os funcionários têm uma escolha de plataformas e dispositivos.

Seguindo a abordagem nativa, isso significava que você precisava construir aplicativos para cada plataforma móvel e, às vezes, aplicativos específicos para *tablets* e *smartphones*.

Com um *framework* híbrido como o *Ionic*, você pode executar seu aplicativo em qualquer plataforma ou dispositivo, tudo a partir de uma única base de código, porque a tecnologia híbrida é baseada na *web*.

Trata-se de um *framework* criado no final de 2013 que visa a criação de aplicações híbridas para dispositivos móveis. Atualmente encontra-se na versão 3, que já passou por diversas melhorias ao longo dos anos. (TABLELESS, 2015)

Ele nada mais é do que uma pilha de componentes e outros frameworks.

Estes componentes são:

### 2.2.1 Node

Com a utilização de sistemas de informação de forma cada vez mais frequente, as aplicações web começaram a ficar com gargalos de tempo entre uma operação e outra, fazendo com que houvesse um *upgrade* nos *hardwares* dos servidores. O node foi desenvolvido com a ideia de um *software* que faz bom uso do *hardware* existente, fazendo com que utilizem o máximo de poder do processador, não o mantendo ocioso. Esta tecnologia foi desenvolvida no final de 2009, por Ryan Dahl e mais 14 colaboradores.

Sendo assim, usuários de sistemas que utilizam a tecnologia Node, estão livres de aguardarem por muito tempo o resultado do processo, fazendo com que a experiência ao usar o aplicativo seja muito mais positiva. (PEREIRA, 2014)

### 2.2.2 Cordova

O Apache Cordova é uma estrutura livre e de código aberto para a construção de aplicativos nativos de plataforma usando HTML, CSS e *JavaScript*.

O benefício principal do Apache Cordova é que ele proporciona a utilização de um conjunto de APIs que se estendem recursos do dispositivo nativo (como a câmera, o acelerômetro, o aplicativo de contatos e assim por diante) para um web aplicativo em execução no contêiner nativo, fazendo com que um aplicativo híbrido possa ter acesso aos recursos do celular, mesmo que de forma mais limitada. (WARGO, 2014, tradução nossa)

Esta tecnologia utiliza um componente chamado *WebView* que funciona como um navegador, mas sem aquelas barras e botões convencionais, fazendo com que se pareça a um aplicativo nativo.

### 2.2.3 Html

As páginas de sites da internet são construídas através da estrutura da linguagem HTML (*HyperText Markup Language*) que é uma linguagem de marcação de documentos com hipertexto. Foi desenvolvida por Tim Berners-Lee em 1980, no início do advento da internet.

Funcionam como se fossem sinais de trânsito para uma página da *web*, cuja função é enviar para o navegador instalado no cliente (Internet Explorer, Firefox, Google Chrome) comandos de formatação que informam ao navegador de que maneira textos e imagens devem ser exibidos na tela (OLIVIERO, 2007, p. 41).

#### 2.2.4 Css

Todo o estilo e formatação das páginas web são definidas pelo CSS, encontra-se nele os mais variados comandos para configuração de cores, tamanhos, fontes, espaçamentos, efeitos. Tornando-se essencial para o desenvolvimento deste tipo de sistema.

Para Macedo (2006) o CSS (*Cascading Style Sheets*) é definido como:

[...] padrão de formatação para documentos HTML/XHTML. Ele permite uma maior versatilidade no desenvolvimento de *design* de sites sem aumentar seu tamanho. Basicamente, o CSS permite ao designer um controle maior sobre os atributos tipográficos de um site, como tamanho e a cor das fontes, espaçamento entre linhas e caracteres (MACEDO, 2006, p.4).

#### 2.2.5 Javascript

JavaScript é uma linguagem de programação criada em 1995 por Brendan Eich, que segundo Silva (2015) tinha o propósito de oferecer aos desenvolvedores formas de tornar determinados processos de páginas web mais dinâmicos, tornando seu uso mais agradável. O funcionamento do *Javascript* consiste em rodar programas localmente - do lado do cliente, fornecendo às páginas web a possibilidade de programação, transformação e processamento de dados enviados e recebidos, interagindo com a marcação e exibição de conteúdo da linguagem *HTML* e com a estilização desse conteúdo proporcionada pelo *CSS* nessas páginas.

#### 2.2.6 Typescript

Trata-se de uma tecnologia que advém do *JavaScript*, criada pela Microsoft, porém com foco maior para desenvolvimento de aplicativos, esta linguagem suporta alguns recursos que o *JavaScript* não suportava, como programação orientada a objetos.

Segundo Npm (2018), o *typescript* é:

[..]uma linguagem para *JavaScript* em escala de aplicativo. O *TypeScript* adiciona tipos opcionais ao *JavaScript* que suportam ferramentas para aplicativos *JavaScript* de grande escala para qualquer navegador, para

qualquer host, em qualquer sistema operacional. O *TypeScript* compila para *JavaScript* legível e baseado em padrões.

### 2.2.7 Angular

Desenvolvido pela Google e parceiros, em 2016 o Angular foi lançado, trata-se de uma plataforma de aplicações web de código aberto, baseado em *javascript* e *typescript*.

Para Portal Gsti (2018) seu objetivo é aumentar aplicativos que podem ser acessados por um navegador web e tem como padrão o *MVVM (Model-View-View-Model)*, em um esforço para facilitar tanto o desenvolvimento quanto o teste dos aplicativos.

## 2.3 Vantagens e desvantagens da utilização do *Ionic framework*

Segundo Ionic (2018) o site oficial, a ferramenta possui atualmente 4 milhões de aplicativos construídos e 5 milhões de desenvolvedores usando a plataforma, o *Ionic* é a plataforma de escolha para qualquer organização que procura desenvolver aplicativos bonitos que fornecem um usuário rico experiência com investimento mínimo.

Dentre as vantagens que o *framework* apresenta pode-se citar como principais:

- *Framework* de código aberto e poderoso conjunto de ferramentas e serviços do desenvolvedor, onde você pode executar seu aplicativo em qualquer plataforma ou dispositivo, tudo a partir de um único código.
- Continuidade de plataforma, a aparência do seu aplicativo não é de tamanho único.
- Tecnologia é toda baseada na *web*, você pode até executar o seu aplicativo em um navegador de *desktop* como um aplicativo da *web*.
- A comunidade de desenvolvedores da *web* é cerca de 30 vezes maior em tamanho do que o número de desenvolvedores de aplicativos móveis nativos.

Muitas equipes de desenvolvimento já têm programadores que entendem de *HTML*, *CSS* e *JavaScript*, tornando-se muito mais fácil recrutar, treinar e contratar especialistas.

- Os usuários têm uma ótima experiência através de plataformas e dispositivos, e o desenvolvedor tem um único código para se preocupar.
- Velocidade, construir para múltiplas plataformas a partir de uma única base de código, o processo é geralmente 2 a 3 vezes mais rápido do que o nativo.
- Eficiência, estima-se que o híbrido pode diminuir de 75-80% em custos de suporte e portabilidade em comparação com o nativo.
- Aplicativos híbridos são capazes de acessar quase todos os recursos nativos de um dispositivo, como a câmera ou o giroscópio, usando *plugins* nativos. O uso desses *plugins* não adiciona complexidade ao desenvolvimento.

A ferramenta pode ser usada em sua versão gratuita que possui algumas limitações, mas também possuem os planos que podem ser adquiridos por desenvolvedores, equipes e empresas.

Com relação às desvantagens pode-se citar como principais:

- As tecnologias utilizadas pelo *Ionic* não permitirão 100% da utilização dos recursos do celular, fazendo com que se o usuário necessitar de maior desempenho dos recursos nativos do *hardware* como câmera, gps, acelerômetro, o *Ionic* não conseguirá atender esta demanda, assim como qualquer outro *framework* de desenvolvimento híbrido, fazendo com que o mesmo deva ser desenvolvido de maneira nativa, ou seja, na linguagem do seu sistema operacional. Para jogos 3D por exemplo e outros apps que necessitam de desempenho, o híbrido pode não ser a melhor escolha.
- Velocidade também pode-se considerar uma desvantagem, pois os aplicativos híbridos também tendem a ser mais lentos se comparado aos nativos. Isso é importante quando os desenvolvedores estão tentando construir aplicações gráficas, como jogos e outros aplicativos de animação intensiva.
- O uso de *SDKs* nativos permite que o desenvolvedor acesse os recursos mais recentes projetados especificamente para essas plataformas, sem a complexidade de lidar com *plugins* nativos. Isso é fundamental quando você

precisa fornecer ao usuário recursos como reconhecimento facial para iOS ou ID de toque para Android, funcionalidade que não pode-se implementar em *apps* híbridos.

### 3 CONCLUSÃO

É fato que a utilização de aplicativos para *smartphones* vem crescendo ao longo do tempo, atualmente se utilizam aplicativos para transporte, alimentação, compras, redes sociais, finanças entre outros.

Através deste estudo, com auxílio de diversos autores aqui citados, pôde-se conhecer sobre esta poderosa ferramenta que vem crescendo no mercado, além de conhecer seu método de funcionamento, quais tecnologias se utilizam, vantagens e desvantagens da utilização da mesma.

Neste artigo foi citado como exemplo o *Ionic Framework*, mas existem diversos de outros *frameworks* de desenvolvimento híbrido. Pôde-se perceber como a ferramenta *Ionic* vem evoluindo, pois trata-se de uma tecnologia relativamente nova que atualmente já possui uma vasta comunidade de desenvolvedores e usuários.

Apesar de pouco tempo no mercado, esta ferramenta já está ajudando muitas pessoas e empresas a satisfazerem as demandas de seus clientes.

Pela ótima estratégia de desenvolvimento e boa aceitação dos usuários, a ferramenta ainda passará por diversas atualizações para facilitar ainda mais a vida dos desenvolvedores, tornando o mercado de tecnologia cada vez mais competitivo e propício a investimentos e pesquisas futuras.

### REFERÊNCIAS

BATTISTELLI, Juliana. **Aplicativos híbridos x aplicativos nativos**. Disponível em: <<https://blog.mastertech.com.br/tecnologia/aplicativos-hibridos-x-aplicativos-nativos/>>. Acesso em: 16 nov. 2018.

IONIC. **A little about Ionic**. 2018. Disponível em: <<https://ionicframework.com/>>. Acesso em: 23 out. 2018.

LAZARO JUNIOR,. **5 vantagens do Ionic para desenvolver suas aplicações mobile**. 2016. Disponível em: <<http://blog.alura.com.br/5-vantagens-do-ionic-para-desenvolver-suas-aplicacoes-mobile/>>. Acesso em: 29 set. 2018.

MACEDO, Marcelo da Silva. **CSS (Folhas de Estilos) – Dicas & Truques**. Rio de Janeiro: Editora Ciência Moderna Ltda 2006.

NPM. TypeScript. 2018. Disponível em: <<https://www.npmjs.com/package/typescript>>. Acesso em: 08 out. 2018.

OLIVIERO, Carlos A. J. **Faça um Site HTML 4.0: Conceitos e Aplicações**. São Paulo: Érica, 2007.

PEREIRA, Caio Ribeiro. **Node.js: Aplicações web real-time com Node.js**. São Paulo: Casa do Código, 2014. 135 p.

PORTAL GSTI. **O que é AngularJS?** Disponível em: <<https://www.portalgsti.com.br/angularjs/sobre/>>. Acesso em: 08 out. 2018.

RANK MY APP. **Aplicativos híbridos x aplicativos nativos**. Disponível em: <<https://www.rankmyapp.com/pt-br/mercado/o-que-sao-progressive-web-apps-e-como-eles-funcionam/>>. Acesso em: 16 nov. 2018.

RUSSELL, Alex. **Progressive Web Apps: Escaping Tabs Without Losing Our Soul**. 2015. Disponível em: <<https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>>. Acesso em: 01 nov. 2018.

SILVA, Giancarlo. **O que é e como funciona a linguagem JavaScript?** 2015. Disponível em: <<https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>>. Acesso em: 08 out. 2018.

SILVA, J. (2015). Desenvolvendo apps multiplataforma para dispositivos moveis com html5. InfoQ.com. Disponível em: [www.infoq.com/br/presentations/desenvolvendoapps-multiplataforma](http://www.infoq.com/br/presentations/desenvolvendoapps-multiplataforma) acesso em: 01 maio 2016.

TABLELESS. **Introdução ao Ionic Framework**. 2015. Disponível em: <<https://tableless.com.br/introducao-ao-ionic-framework/>>. Acesso em: 23 out. 2018.

WARGO, John M. **Apache Cordova 3: Programming**. New Jersey: Pearson Education, Inc., 2014. 263 p.