INTERFACE TECNOLÓGICA

A UTILIZAÇÃO DO IONIC FRAMEWORK NO DESENVOLVIMENTO DE APLICAÇÕES HÍBRIDAS EM ARQUITETURA ORIANTADA A SERVIÇO

USING THE IONIC FRAMEWORK IN THE DEVELOPMENT OF HYBRID APPLICATIONS IN SERVICE ORIENTED ARCHITECTURE

Ewerthon Patricio Araujo da Silva – ewerthon.patricio@gmail.com
Eder Carlos Salazar Sotto - eder.sotto@fatectq.edu.br
Faculdade de Tecnologia de Taquaritinga (FATEC) –SP –Brasil

DOI: 10.31510/infa.v15i1.333

RESUMO

Este artigo foi idealizado e motivado pela crescente utilização e desenvolvimento de aplicações móveis, aplicações com arquitetura orientada a serviço e a utilização do Ionic *Framework* para desenvolvimento de aplicações híbridas, com o objetivo de elucidar e descrever as principais características do Ionic *Framework*, da Arquitetura Orientada a Serviços, além das dificuldades e desafios desse tipo de desenvolvimento. Para isso, foram realizadas pesquisa científica em artigos, fóruns e revistas especializadas. O resultado obtido foi uma melhor compreensão e orientação sobre quais cuidados deve-se tomar para o desenvolvimento deste tipo de aplicação, levando em consideração que o Ionic *Framework* possui uma vasta quantidade de recursos e uma grande comunidade ativa, além de boa documentação, trazendo inúmeros benefícios a seus desenvolvedores. Com as devidas análises e pesquisas é possível concluir que é extremamente viável o uso do Ionic Framework no desenvolvimento de aplicações hibridas em arquiteturas orientadas a serviço.

Palavras-chave: Ionic. Arquitetura Orientada a Serviço. Desenvolvimento. Aplicações Híbridas. SOA.

ABSTRACT

This article was conceived and motivated by the growing use and development of mobile applications, applications with service oriented architecture and the use of the Ionic Framework for the development of hybrid applications, with the objective of elucidating and describing the main features of the Ionic Framework, Service Oriented Architecture, and the difficulties and challenges of this type of development. For this, scientific research was carried out in articles, forums and specialized magazines. The result obtained was a better understanding and guidance on what care should be taken to develop this type of application, assuming that the Ionic Framework has a large amount of resources and a large active community, as well as good documentation, bringing numerous benefits to its developers. With the proper analysis and research, it is possible to conclude that the use of the Ionic Framework in the development of hybrid applications in service oriented architectures is extremely viable.

Keywords: Ionic. Service Oriented Architecture. Development. Hybrid Application. SOA.

1 INTRODUÇÃO

Com o aumento crescente da quantidade de dispositivos móveis no mercado e o consequente aumento da demanda por aplicações móveis, surgiu a necessidade de desenvolvimento rápido de aplicativos para plataformas distintas, algo considerado caro e que demanda muito tempo, visto que cada plataforma tem um ambiente de desenvolvimento completamente diferente das demais. O desenvolvimento de aplicativos híbridos possibilitou a criação de um novo paradigma. Isso pode ser atribuído à possibilidade do desenvolvimento em menor tempo, devido principalmente a reutilização do código para diversas plataformas (PREZOTTO; BONIATI apud MATOS; DE BRITTO E SILVA, 2016).

1.1 Objetivo

O objetivo deste trabalho é definir, por meio de pesquisa, as principais características do Ionic *Framework*, do Desenvolvimento Híbrido e da Arquitetura Orientada a Serviço, além das dificuldades e desafios desse tipo de desenvolvimento, explanando sobre o que é o desenvolvimento híbrido, quais precauções deve-se tomar ao optar pelo desenvolvimento deste tipo de aplicativo, além de apresentar as principais vantagens e desvantagens do uso do Ionic *Framework*.

2 REFERENCIAL TEÓRICO

Neste Capítulo é apresentada uma pesquisa bibliográfica referente a fundamentação teórica sobre o tema abordado no presente artigo.

2.1 Tecnologias

A seguir são apresentados os conceitos e tecnologias utilizados no desenvolvimento deste artigo.

2.1.1 Desenvolvimento Híbrido

De acordo com PONTOCODE (2017):

> O desenvolvimento híbrido é uma modalidade de desenvolvimento que utiliza as tecnologias *Web* como HTML5, *Javascript* e CSS, em conjunto com algum *framework* que tenha acesso às funções nativas do aparelho, como por exemplo GPS, câmera e sensores, com a finalidade de possibilitar a criação de aplicações de forma mais fácil, pois utiliza de tecnologias com maior visibilidade e utilização pela comunidade de desenvolvimento.

Nos últimos anos o desenvolvimento de aplicativos híbridos tem crescido, principalmente devido ao fato de sua linha de aprendizagem ser menor que o aprendizado para desenvolvimento de um aplicativo nativo. (LYNCH, 2017)

Para Korf e Oksman (2016), o desenvolvimento híbrido se caracteriza pela criação do aplicativo como uma aplicação Web, que foi construída, primeiramente, utilizando o HTML5 e o *JavaScript.* Após esta etapa, são adicionadas bibliotecas que fornecem acesso a recursos da plataforma nativa.

Os aplicativos híbridos têm muitos benefícios em relação aos aplicativos nativos puros, especificamente em termos de suporte à plataforma, velocidade de desenvolvimento e acesso ao código de terceiros. (IONIC FRAMEWORK, 2016)

2.1.2 Ionic Framework

Ionic é um *framework* de desenvolvimento de aplicativos móveis em HTML5 voltado para o desenvolvimento híbrido. Ele utiliza o Angular, que é um *framework JavaScript* desenvolvido pela Google. Os aplicativos desenvolvidos com Ionic são como pequenos sites que funcionam em um uma camada embutida do navegador em um aplicativo, o qual possui acesso à camada da plataforma nativa através de bibliotecas fornecidas pelo próprio Ionic.

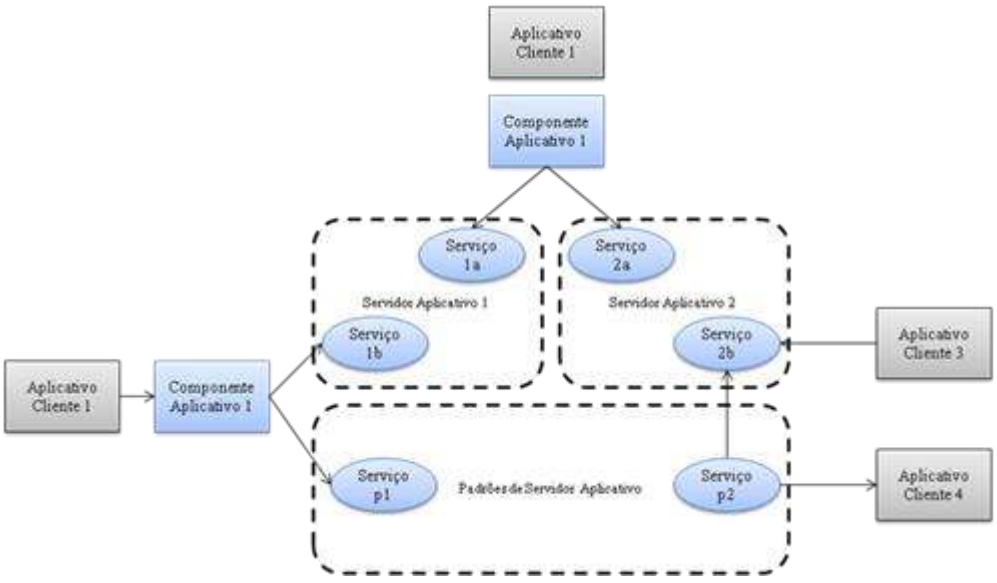
Ao contrário de uma estrutura responsiva qualquer, o Ionic vem com elementos e *layouts* de UI móveis de estilo nativo, que só seria possível obter com um SDK nativo no iOS ou no Android, mas que não seria possível nos elementos e *layouts* da *Web*. (IONIC FRAMEWORK, 2016).

Por se tratar de um *framework* HTML5, o Ionic precisa de um invólucro nativo como o Cordova ou PhoneGap para ser executado como um aplicativo nativo, assim como acessar algumas funcionalidades nativas do equipamento.

2.1.3 Arquitetura Orientada a Serviço

Segundo Furtado *et al.* (2009), Arquitetura Orientada a Serviço ou *Service Oriented Architecture* (SOA) é um modelo de Projeto que tem como objetivo orientar e guiar todos os aspectos da criação e do uso de serviços de negócio por todo o ciclo de vida de desenvolvimento, que seria desde a concepção da ideia até o momento em que o serviço é descontinuado. A SOA também trata das definições e dos provisionamentos da parte de infraestrutura de TI, que permite que diferentes aplicações troquem dados e se comuniquem indiferentemente dos sistemas operacionais, aparelhos ou locais onde estas aplicações estejam sendo executadas, ou até mesmo da linguagem de programação que foi utilizada para o desenvolvimento das mesmas, como pode ser observado na Ilustração 1. (FURTADO et. Al., 2009).

Ilustração 1 - Diagrama de funcionamento de aplicação em SOA



Fonte: MARIOTTI (2017)

SOA é uma abordagem de negócios para criar sistemas de Tecnologia de Informação (TI) que permitem alavancar recursos existentes, criar novos recursos e, principalmente,

possibilitar as melhorias e inevitáveis alterações exigidas pelo mercado, obtendo mais produtividade e lucro para a empresa (BENTO, 2012).

Arquitetura Orientada a Serviços é um modelo de planejamento de estratégia da área de tecnologia da informação, alinhando diretamente aos objetivos de negócios de uma organização. Esta ponte permite expor as funcionalidades dos aplicativos em serviços padronizados e inter-relacionados. (AVELLAR E DUARTE, 2012 apud MENDES, 2016)

2.2 Utilização do Ionic *Framework* com SOA

Nesta subseção são apresentadas vantagens e desvantagens do uso do Ionic *Framework,* além dos cuidados que devem ser tomados neste tipo de desenvolvimento.

2.2.1 Vantagens do uso do Ionic *Framework*

Segundo Kumar (2017), Patel (2017) e Blini (2016) os motivos que incentivaram a escolha do Ionic *Framework* como seu *framework* para criação de aplicações hibridas são:

- Ser completamente gratuito;
- Ter como base o Angular, *framework* que está sendo desenvolvido e mantido pelo Google, em licença do tipo *Open Source*;
- Os aplicativos desenvolvidos com o Ionic se comportam como aplicativos nativos;
- Possuir suporte a diversas Plataformas como *Android e* iOS;
- Possuir curva de aprendizagem muito pequena, permitindo a construção de um aplicativo com pouco esforço e tempo, considerando prévio conhecimento em HTML, CSS e Javascript;
- Possibilidade de visualizar o aplicativo diretamente no navegador durante o desenvolvimento, sem a necessidade de emular um dispositivo nativo. O *Chrome* e o *Firefox* possuem excelentes Ferramentas de Desenvolvimento para visualizar e testar aplicativos em desenvolvimento;
- Possuir manutenção de código centralizada, já que o código que roda no *Android* é o mesmo que roda no iOS e outras plataformas;

- Ter suporte aos *Plugins* do Cordova e do Ionic *Native Plugins,* além dos Módulos do NodeJs;
- Possuir uma comunidade grande e ativa;
- Possuir diversas ferramentas de apoio ao desenvolvedor como, por exemplo, o *Ionic Creator, Ionic Lab, Ionic Platform, Ionic Playground e Ionic View App*

De acordo com Pereira Silva e Deus Lima (2016), a curva de aprendizagem para a utilização do Ionic *Framework* é considerada baixa, pois desenvolvedores que possuem domínio sobre as linguagens utilizadas já saem na frente no processo de aprendizagem.

Além dos motivos acima, podemos citar também que o Ionic é um *framework* que começou a ser desenvolvido em 2013 e atualmente já está na sua versão 3.9, onde após a sua versão 2.X passou por uma reformulação total, deixando-o melhor estruturado e com uma melhor documentação.

Segundo Kumar (2017), o Ionic é o *framework* de desenvolvimento de aplicações móveis híbridas dominante no mercado de aplicações híbridas e está bem posicionado para continuar com sua predominância e tornar-se ainda melhor no futuro.

2.2.2 Desvantagens do uso do Ionic *Framework*

Para Blini (2016), as desvantagens de se desenvolver utilizando o Ionic são:

- Problemas de performance em alguns ambientes, principalmente em Androids mais antigos;
- Pode apresentar discrepâncias entre componentes UI entre o código desenvolvido e o que é renderizado na plataforma nativa;
- Não é seguro que funcione bem em 100% dos dispositivos
- Pode apresentar problemas de performance, quando necessita de grandes recursos gráficos

2.2.3 Precauções

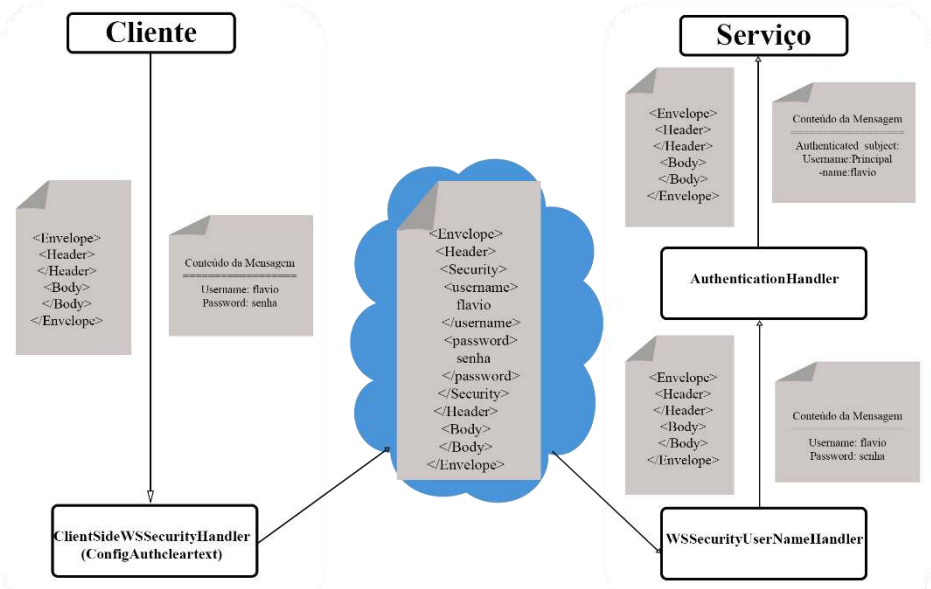
Quando se fala em desenvolver aplicações que irão consumir serviços, deve-se tomar algumas precauções para que não coloquemos, nem o aparelho do cliente, nem o serviço, nem os dados trafegados em risco. Para que isso não ocorra, é importante consultar no site do SOA

Patterns informações sobre os padrões de boas práticas para o desenvolvimento de aplicações no padrão SOA.

Segundo Mariotti (2017), os principais pontos da segurança em aplicações SOA são:

- Blindagem de exceções;
- Utilização de recursos do *WS-Security* (Segurança em Serviços Web) no protocolo de transferência SOAP (*Simple Object Access Protocol*), definindo os cabeçalhos de segurança padrão para o SOAP, conforme demonstrado na Ilustração 2;
- Política de segurança para Codificação;
- Segurança declarativa;
- Confidencialidade das mensagens usando criptografia.

Ilustração 2 - Visão geral da implementação de um nome de usuário e senha



Fonte: Adaptado de MARIOTTI (2017)

Para Groffe (2016) devemos ter os seguintes cuidados ao utilizar SOA:

- A equipe envolvida deve ser capacitada e estar familiarizada com conceitos desta arquitetura;
- Mudanças drásticas em um serviço podem resultar em efeitos indesejáveis em outras aplicações;
- Questões envolvendo a segurança na transmissão de informações também devem ser priorizadas;

- Analisar a real necessidade de implementação de um serviço (potencial de reuso, questões de infraestrutura) é um aspecto de fundamental importância, sobretudo por evitar esforços que podem se revelar como infrutíferos.

Além dos cuidados de se desenvolver para arquiteturas SOA, deve-se também tomar algumas precauções em relação ao desenvolvimento com o Ionic, ainda mais depois da versão 2.X, onde houve uma reestruturação na sua estrutura de arquivos, que passou a adotar a estrutura do Angular, a qual podemos verificar no Guia de Estilo do Angular em https://angular.io/guide/styleguid. Abaixo alguns dos cuidados que devem ser respeitados (ANGULAR, 2016).

- Definir um componente por arquivo – como por exemplo o ngRoute, *Controller* e o *Factory* devem estar em arquivos separados;
- Envolver seus componentes AngularJS em uma "*Immediately Invoked Function Expression*" - IIFE.
- Criar uma diretiva por arquivo, com o mesmo nome.
- Fornecer um prefixo curto, único e descritivo, como acmeSalesCustomerInfo que é declarado no HTML como acme-sales-customer-info.
- Utilizar controllers ao invés do modo clássico com $scope.
- Utilizar uma variável de captura quando estiver usando controller.
- Definir um nome de variável consistente como "vm", que representa o padrão *ViewModel*.

2.3 Exemplos de Apps Ionic *Framework*

A seguir são apresentados alguns exemplos de aplicativos desenvolvidos com Ionic *Framework* em SOA. Além dos exemplos apresentados, existem outros que podem ser consultados em http://showcase.ionicframework.com/apps/top (IONIC FRAMEWORK, 2017).

- *Ionic View* – É um aplicativo que permite testar e compartilhar seus projetos em Ionic com colaboradores, clientes, entre outros;
- TD *Trading* – É um aplicativo que oferece serviços de investimentos como por exemplo ações, fundos de investimento;

- *Nationwide* – Esse aplicativo oferece serviços de pagamentos, consultas e contato com agente bancário para clientes da *Nationwide*®

3 PROCEDIMENTOS METODOLÓGICOS

Visando identificar as principais características, definições, vantagens, desvantagens e precauções para o desenvolvimento de aplicações híbridas em arquitetura orientada a serviço com a utilização do Ionic *Framework*, a análise sistemática apresentada nesse artigo foi concluída segundo os tópicos a seguir, a saber:

- Levantamento bibliográfico acerca dessas ferramentas;
- Análise teórica de cada processo inerente a essas ferramentas;
- Listagem de opiniões dos autores sobre o assunto abordado;
- Confrontamento dos conceitos levantados;
- Avaliação de resultados.

Inicialmente, foi feito o levantamento de autores reconhecidos por seu conhecimento na área de desenvolvimento de aplicativos mobile, e suas respectivas obras.

Posteriormente, foi realizada uma exploração bibliográfica a respeito dos itens selecionados, salientando seu funcionamento, suas principais características e tarefas e qual o propósito de utilizá-lo.

Concluindo essa pesquisa, o artigo apresenta a análise detalhada dessa modalidade de desenvolvimento, da ferramenta, bem como da arquitetura abordada.

4 RESULTADOS

Mediante as pesquisas realizadas conseguimos constatar que, apesar das desvantagens, a utilização do Ionic *Framework* para desenvolvimento de aplicativos híbridos mobile em SOA é extremamente viável, principalmente devido a sua menor curva de aprendizagem e tempo de desenvolvimento, o que acaba gerando menos custos. O gasto para desenvolver um mesmo aplicativo para diferentes plataformas pode ser alto e inviabilizar o projeto. O fato de muitos desenvolvedores já possuírem domínio, ou até mesmo algum conhecimento sobre as linguagens para Web utilizadas também acabam contribuindo para uma menor curva de aprendizado, além

do fato dos desenvolvedores contarem com uma com uma comunidade grande e ativa, e uma documentação que é bem estruturada e explicativa.

5 CONSIDERAÇÕES FINAIS

Com as devidas análises e pesquisas é possível concluir que é extremamente viável o uso do Ionic *Framework* no desenvolvimento de aplicações hibridas em arquiteturas orientadas a serviço, por se tratar de um *framework* robusto e que possui uma grande comunidade ativa e uma boa documentação, além de utilizar código único para o desenvolvimento e manutenção dos aplicativos, diminuindo os custos para desenvolvimento dos projetos. No entanto, é importante ressaltar que o sucesso ou fracasso no desenvolvimento de aplicações dependerá de que sejam tomadas as devidas precauções como as citadas no item 2.2.3 deste artigo, além da realização da devida análise de requisitos, para que assim possam ser reduzidos os riscos e aumentadas as chances de sucesso do projeto.

REFERÊNCIAS

ANGULAR, **Angular – Style Guide**. Disponível em <https://angular.io/guide/styleguide>. Acesso em: 21 ago. 2017.

BENTO, Rafael. **Introdução a Arquitetura orientada a serviços (SOA)**. Disponível em: <http://www.devmedia.com.br/introducao-a-arquitetura-orientada-a-servicos-soa/27260>. Acesso em: 22 jun. 2017.

BLINI, Felipe. **Desenvolvimento de Aplicativos Móveis com Javascript: Ionic, React Native e NativeScript. Qual escolher?**. Disponível em: <https://medium.com/@felipe blini/desenvolvimento-de-aplicativos-m%C3%B3veis-com-javascript-ionic-react-native-e-nativescript-c303b17fba0d>. Acesso em: 21 ago. 2017.

FURTADO, Camille et al. **Arquitetura Orientada a Serviço - Conceituação**. 2009. 9 p. Relatórios Técnicos do Departamento de Informática Aplicada - Departamento de Informática Aplicada, Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro – Brasil, 2009. Disponível em: <https://www.researchgate.net/profile/Fernanda_Baiao/publication/277237476_00122009_-_Arquitetura_Orientada_a_Servico_-_Conceituacao/links/558c477e08aee43bf6ae289a/0012-2009-Arquitetura-Orientada-a-Servico-Conceituacao.pdf>. Acesso em: 26 jun. 2017.

GROFFE, Renato. **Arquitetura Orientada a Serviços (SOA): uma visão geral**. Disponível em: <http://netcoders.com.br/arquitetura-orientada-a-servicos-soa/>. Acesso em: 25 ago. 2017.

IONIC, **Ionic Framework - Preface**. Disponível em <http://ionicframework.com/docs/v1/guide/preface.html>. Acesso em: 25 ago. 2017.

KORF, Mario; OKSMAN, Eugene. **Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options**. Disponível em: <https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options>. Acesso em: 25 maio 2017.

KUMAR, Ajay. **Top 10 Reasons to choose Ionic framework for mobile app Development**. Disponível em: <http://www.techexceed.com/reasons-to-choose-ionic-framework-for-mobile-app-development/>. Acesso em: 22 ago. 2017.

LYNCH, Max. **Hybrid apps are unstoppable.** Disponível em: < https://medium.com/ionic-and-the-mobile-web/hybrid-apps-are-unstoppable-a113b0b413dc> Acesso em: 20 ago. 2017

MARIOTTI, Flávio S. **Segurança em SOA**. Disponível em: <http://www.linhadecodigo.com.br/artigo/3076/seguranca-em-soa.aspx>. Acesso em: 22 ago. 2017.

MATOS, Beatriz R D; DE BRITTO E SILVA, João G. **Estudo comparativo entre o desenvolvimento de aplicativos móveis utilizando plataformas nativas e multiplataforma**. 2016. Trabalho de Conclusão de Curso - Monografia (Título de Bacharel em Engenharia de Software) - Faculdade UnB Gama, Universidade de Brasília, Brasília, DF, 2016. Disponível em: <https://fga.unb.br/articles/0001/5113/Beatriz_Joao_TCC_Aplicativos_M_veis.pdf>. Acesso em: 21 ago. 2017.

MENDES, Edson. **Vantagens e Desvantagens de SOA**. Disponível em: <http://www.devmedia.com.br/vantagens-e-desvantagens-de-soa/27437>. Acesso em: 28 jul. 2017.

PATEL, Ronak. **7 Reasons to Choose Ionic Framework for your next Mobile App**. Disponível em: <https://www.linkedin.com/pulse/7-reasons-choose-ionic-framework-your-next-mobile-app-ronak-patel>. Acesso em: 22 ago. 2017.

PEREIRA SILVA, Vinícius; DE DEUS LIMA, Dênis Henrique. **Desenvolvimento De Aplicativos Híbridos Benefícios E Problemas Encontrados Na Utilização Dessas Tecnologias.** 2016. 6 f. (Técnico em Eletrônica Integrado ao Ensino Médio) - Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro, Minas Gerais - Brasil, 2016. Disponível em: <http://www.ifetec.com.br/evento/index.php/enpe/3enpe/paper/viewFile/151/78>. Acesso em: 22 ago. 2017.

PONTOCODE. **Você sabe o que é Desenvolvimento Híbrido?**. Disponível em: <http://pontocode.com.br/novidades/voce-sabe-o-que-e-desenvolvimento-hibrido/>. Acesso em: 26 jun. 2017.

RAJPUT, Mehul. **Why you should choose Ionic Framework to build your next Mobile App?**. Disponível em: <http://www.mindinventory.com/blog/why-you-should-choose-ionic-framework-to-build-your-next-mobile-app/>. Acesso em: 22 ago. 2017.

ROUSE, Margaret. **Definition: Hybrid Application (hybrid app)**. Disponível em: <http://searchsoftwarequality.techtarget.com/definition/hybrid-application-hybrid-app>. Acesso em: 22 jun. 2017.