Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

ANÁLISE TEÓRICA SOBRE O DESENVOLVIMENTO DE APLICATIVOS NATIVOS, HÍBRIDOS E WEBAPPS

Fransérgio Aparecido de Souza Silva[1]

Me. Ely Fernando do Prado[2]

Resumo

O mercado de aplicativos para dispositivos móveis vive um impasse nos dias atuais, consideráveis diferenças entre os principais sistemas operacionais para dispositivos móveis, Android e iOS, incluindo grandes diferenças nas linguagens de programação utilizadas para cada plataforma, sem falar nos ambientes de desenvolvimento. Um comportamento crescente a fim de atenuar este problema é o desenvolvimento de aplicações híbridas e aplicações Web que compilam o mesmo código para ambas as plataformas. O presente trabalho tem por objetivo apresentar um estudo teórico sobre as tecnologias nesta esfera de desenvolvimento. Para tal, foram analisados artigos acadêmicos, livros e documentação sobre o desenvolvimento dos tipos de aplicações em questão. O estudo sobre cada técnica de desenvolvimento foi feito mostrando as características de cada tipo de aplicação, linguagens utilizadas e vantagens e desvantagens de cada um. Ao final do estudo pode se concluir que os desenvolvedores dependendo de suas necessidades podem optar por qual tipo de desenvolvimento lhe atende de melhor forma.

Palavras-chave: Android. Desenvolvimento. Dispositivo móvel. Híbridas. iOS

Abstract

The mobile application (app) market is experiencing a deadlock nowadays, considerable differences between the major operating systems for mobile devices, Android and iOS, including large differences in programming languages used for each platform, not to mention development environments. Increasing behavior to mitigate this problem is the development of hybrid applications and Web applications that compile the same code for both platforms. This paper aims at presenting a theoretical study about technology in this sphere of development. Thus, we analyzed academic articles, books and documentation on the development of the types of applications in question. The study on each development technique was done showing the characteristics of each type of application, languages used and advantages and disadvantages of each one. At the end of the study it can be concluded that developers depending on their needs can choose which type of development suits them best.

Keywords: *Android. Development. Mobile device. Hybrids. iOS*

[1] Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: fransergioss@hotmail.com.
[2] Docente da Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: elyfprado@gmail.com

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

1 Introdução

Em 2022 serão feitos 258,15 bilhões de *downloads* de *apps* no mundo, crescimento médio anual de 7,7% em cinco anos segundo o site Mobile Time[3] (2018). Diante de tal dado é valido considerar que que várias empresas vêm buscando oferecer aos seus clientes a facilidade de utilizar seus serviços com o conforto e comodidade de suas casas através de um smartphone, *tablet*, ou outro dispositivo móvel conectado à Internet.

Com o crescimento de serviços oferecidos pela internet e o número cada vez maior de usuários de dispositivos móveis, ocorre uma corrida pelo desenvolvimento de aplicativos para sistemas operacionais móveis, sendo este mercado dominado pelo sistema operacional Android com 85,9% dos usuários, seguido do iOS com 14%dos usuários (MACMAGAZINE.UOL, 2018). Para atender a crescente procura por aplicativos as empresas têm a dificuldade de encontrar equipes de desenvolvimento com desenvolvedores habilitados em pelo menos duas linguagens de programação, e com conhecimento prático nas diversas características singulares de cada plataforma para realizar um desenvolvimento nativo. Estas diferenças além de dificultarem o desenvolvimento de aplicações compatíveis com as duas principais plataformas, aumenta a possibilidades de erros e os custos de desenvolvimento para ambas as plataformas separadamente, bem como os custos de manutenção para a correção de bugs em códigos fonte diferentes, mas com um mesmo objetivo final. Por outro lado, o desenvolvimento nativo possibilita o máximo na utilização do hardware do dispositivo.

Segundo Lopes(2016):

> A solução mais comum atualmente para a construção de aplicativos multiplataforma é o Cordova, que é basicamente uma mistura. Ele usa o ponto forte a Web de ter linguagens padronizadas e um ambiente de execução, o navegador, para construir aplicativos.

As aplicações Híbridas também têm se demonstrado como uma alternativa e com um considerável crescimento entre os desenvolvedores, principalmente com o uso de frameworks como o Cordova, Phonegap entre outros. São como websites tradicionais que executam através de um navegador, uma *webview* de um sistema

[3] https://www.mobiletime.com.br/previsoes/em-2022-serao-feitos-25815-bilhoes-de-downloads-de-apps-no-mundo-crescimento-medio-anual-de-77-em-cinco-anos/

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

operacional móvel com uma “roupa” de aplicativo nativo, que pode ser colocada a venda nas lojas de aplicativos, instalada no dispositivo, tem acesso ao hardware do dispositivo com algumas restrições, porém ainda são sites e tem seu desempenho diretamente ligado ao motor de execução web, e normalmente tem um menor desempenho ao comparados com aplicativos nativos.

Com a utilização cada vez maior da plataforma Web, é comum que exista uma série de desenvolvedores web bem capacitados no mercado de maneira que surge como uma alternativa de desenvolvimento são as webapps. Sendo assim são soluções puramente web que se comportam como aplicativos, mas que podem sofrer problemas com o desempenho e acesso as funções do dispositivo, não são instaláveis, mas com as constantes inovações de mercado e com o surgimento da tecnologia PWA (*Progressive Web Apps*) busca-se a cada dia resolver tais problemas de desempenho.

De acordo com Jones (2011, p. 5-6):

> Uma alternativa ao desenvolvimento de aplicativos nativos é criar, em vez disso, um aplicativo da Web. Isso envolve o uso de HTML, JavaScript e CSS mais familiar e fácil de aprender [...]. A distribuição para aplicativos da Web é através de um servidor Web, e não de uma loja de aplicativos [...]. A velocidade de execução provavelmente será mais lenta [...]. Também haverá acesso mais limitado ao hardware do dispositivo.

O objetivo do presente trabalho é um estudo teórico sobre algumas tecnologias neste âmbito de desenvolvimento, através de uma análise de artigos acadêmicos, livros, revistas e documentação das práticas de desenvolvimento apresentadas, afim de expor seus pontos positivos e negativos, bem como linguagens de programação e características de cada tipo de tecnologia apresentada, para que assim possa se fazer uma análise de cada tipo de desenvolvimento e optar pela forma que melhor atende os requisitos de cada aplicação bem como suas necessidades.

2. Dispositivos Móveis

Há um senso comum para caracterizar se um aparelho é um dispositivo móvel, imagina-se que qualquer dispositivo eletrônico e tecnológico com a possibilidade de ser transportado pelo seu usuário é considerado um dispositivo móvel. No entanto na literatura existem definições para o conceito de dispositivo móvel como quando Firtman (2013), destaca que para receber a denominação de

ISSN: 2595-5942

dispositivo móvel, os aparelhos necessitam comtemplar as seguintes características: serem portáteis, podendo ser utilizados pelos usuários em qualquer lugar; ser pessoal devendo ser utilizado apenas para uso pessoal, onde o usuário personaliza o aparelho conforme sua necessidade e gosto; fácil de usar, precisa ser rápido e de fácil acesso; conectado, deve estar pronto para se conectar à internet onde e quando o usuário precisar.

Fitzek e Reichart (2007), apresentam três características que representam as funcionalidades dos dispositivos móveis: a interface com o usuário, através de meios de entrada e saída de dados; a interface de comunicação, apresentando várias formas de acesso à internet, e troca de dados entre os aparelhos por meio de 4G, *wifi*, e *bluetooth;* e os recursos internos, que dizem respeito a estrutura adotada nos dispositivos como a unidade de armazenamento, processamento e bateria.

O mercado de dispositivos móveis sofre constantes mudanças e é por esse motivo que o grande desafio é ter uma previsão do que pode acontecer nos próximos anos já que as mudanças ocorrem em curto prazo. A única certeza que se tem é que a cada dia a tecnologia vai evoluir cada vez mais.

De acordo com Siqueira (2011), a previsão é que haja 5 bilhões de dispositivos móveis no mundo em 2020.

Atualmente os dispositivos móveis vêm sido utilizados nas mais diversas áreas, e cada vez mais essa utilização vem se expandindo, pois há uma evolução natural em que gerações mais antigas tem se apropriado mais a cada dia destas tecnologias, e novas gerações que são chamados “nativos digitais”, já tem esses dispositivos como se fossem parte do vestuário, um utensílio indispensável para o dia-a-dia, assuntos do cotidiano, textos científicos, fotos, namoros, relações de trabalho, não necessitam mais da presença física para que se realizem.

E a grande característica desses dispositivos é precisamente a de serem móveis. Esta característica faz com que em tempo real sejam atualizadas várias informações em redes sociais, em funções de trabalhos, em localização e posicionamento. A junção de recursos como o GPS (*Global Positioning System*) auxilia na mobilidade até mesmo de pessoas com algum tipo de deficiência.

> A tecnologia visto antes como algo que tirava o sujeito do convívio social e do contato coletivo, torna-se cada vez mais customizadora, assim os ambientes tornam-se individualizados, mas não individualistas. Os dispositivos, os aplicativos e suas interfaces podem ser cada vez mais customizados e personalizados. Os ambientes ganham fotos, perfis e avatares criando uma atmosfera mais humanizada, representando um ponto

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

> muito positivo para a pedagogia centrada no aluno (TOTTI, GOMES, MOREIRA, SOUZA, 2011, p2).

2.1 ANDROID

O Android é um sistema operacional que tem sua base no *kernel* do Linux, desenvolvido inicialmente para *smartphones,* mas atualmente é usado em diversos dispositivos como *tablets, netbooks*, relógios e outros dispositivos (GOMES 2012, p7).

Apesar de ter sua base no *kernel* do Linux, pouca coisa existe em comum com as distribuições Linux convencionais embarcadas ou não, lembrando que o sistema embarcado é um sistema microprocessado no qual o computador é encapsulado ao dispositivo que ele controla. De uma maneira rude, o Android é uma máquina virtual Java rodando sobre o *kernel* do Linux, que através de um conjunto de bibliotecas e serviços, que dão suporte ao desenvolvimento de aplicações Java.

Segundo Developers (2011, p 48), o Android é uma pilha de softwares para dispositivos móveis que incluem um sistema operacional, middleware e aplicativos-chave.

Todas as aplicações são escritas em linguagem JAVA, sua língua nativa, então é nesta camada que você poderá criar seu aplicativo móvel. Os desenvolvedores têm acesso total às mesmas API's de *frameworks* usadas pelos principais aplicativos. A arquitetura da aplicação é projetada para simplificar a reutilização de componentes, os recursos de qualquer aplicação podem ser publicados e, em seguida, ser utilizado por qualquer outro aplicativo[4].

Você pode desenvolver aplicativos Android com as mesmas ferramentas de alta qualidade que você usa para desenvolver aplicativos Java. As bibliotecas principais do Android fornecem a funcionalidade necessária para criar algumas aplicações móveis incrivelmente ricas.

O Android conta com o *kernel* do Linux para serviços essenciais do sistema, como segurança, gerenciamento de memória, gerenciamento de processos, pilha de rede e modelo de driver. O *kernel* também atua como uma camada de abstração entre o hardware e o resto da pilha de software. Várias funções do *Kernel* são

[4] Sujeito a restrições de segurança impostas pela estrutura.

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

utilizadas diretamente pelo Android, mas muitas modificações foram feitas para otimizar memória e tempo de processamento dos aplicativos.

2.2 iOS

O iOS (*iPhone Operating System*) tem sua arquitetura formada por quatro camadas, sendo que cada uma delas oferece um conjunto de frameworks que durante o desenvolvimento de aplicativos para dispositivos móveis da Apple podem ser utilizados.

Anvaari and Jansen, (2010) dizem que a arquitetura do iOS é semelhante à arquitetura básica encontrada no Mac OS.

O iOS atua como um intermediário entre o hardware subjacente e os aplicativos que aparecem na tela no nível mais alto.

No iOS os aplicativos desenvolvidos raramente se comunicam diretamente com o *hardware* do dispositivo, ao invés disso, os aplicativos se comunicam com o *hardware* através de um conjunto de interfaces de sistemas bem definidas que protegem seu aplicativo de qualquer alteração do *hardware* (APPLE, 2008).

O iOS foi desenvolvido para atender às necessidades de um ambiente móvel. Desenvolvedores do Mac OS X vão encontrar muitas tecnologias familiares, mas também vão encontrar tecnologias que só estão no iOS (GONZALES-SANCHEZ and CHAVEZ-ECHEAGARAY, 2010).

O iOS é formado por camadas: Core OS, Core Services, Media e Cocoa Touch (YATES, 2010). Nas camadas superiores estão as tecnologias e serviços mais avançados, nesta camada estão os *frameworks* que fornecem abstração orientadas a objetos das camadas de níveis inferiores. Estas abstrações facilitam o desenvolvimento e escrita de código, pois reduzem a quantidade de código que o desenvolvedor tem que escrever, e encapsula características complexas, tais como *threads*.

No sistema os serviços fundamentais e as tecnologias dos quais os aplicativos dependem estão nas camadas inferiores. Embora que as tecnologias nos níveis superiores se resumam a tecnologia de níveis inferiores, os desenvolvedores ainda podem usar essas últimas que não estão presentes nas camadas superiores (APPLE, 2010).

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

Na camada Cocoa Touch estão os principais *frameworks* para a construção de aplicações. É nela que se define a infraestrutura das tecnologias fundamentais, tais como multitarefa, serviço de notificação Apple *push* e diversos serviços do sistema.

Na camada média, estão as tecnologias que foram desenvolvidas para tornar mais fácil a implementação de aplicativos de multimídia como as tecnologias de gráfico, áudio e vídeo.

No nível superior os *frameworks* oferecem tecnologias que facilitam a criação de gráficos e animações, enquanto os *frameworks* de nível inferior permitem o acesso às ferramentas fundamentais para que o desenvolvedor possa utilizar para a criação de aplicativos mais complexos e robustos.

As tecnologias para gravar e reproduzir conteúdo baseados em vídeos também estão disponíveis na camada média.

Os serviços fundamentais do sistema que todos os aplicativos utilizam estão na camada Core Services.

As tecnologias de baixo nível utilizadas para a implementação de outras tecnologias estão na camada Core Os. Quando o desenvolvedor precisa lidar explicitamente com segurança ou comunicação com acessório de *hardware* externo, ele pode fazer isso utilizando os *frameworks* existentes nesta camada.

No ambiente *kernel* a nível de sistema está presente drivers e interfaces de baixo nível do sistema operacional UNIX.

A interface entre o *hardware* disponível e os *frameworks* são fornecidas através de *drivers*, o iOS fornece um conjunto de interfaces para acessar várias características de baixo nível do sistema operacional.

O desenvolvimento de aplicativos móveis vem crescendo constantemente a cada dia e a necessidade de profissionais qualificados é cada vez maior, o iOS atua como intermediário entre o hardware e a aplicação instalada, desta forma os aplicativos desenvolvidos para ele podem rodar em outros dispositivos com iOS garantindo uma maior portabilidade, diferentemente em aplicativos desenvolvidos para Android em smartphones que, quando executados em dispositivos com telas maiores, o layout dos dispositivos é desconfigurado. O iOS foi projetado para executar tanto aplicativos nativos quanto aplicativos web, a vantagem desse sistema é que em regiões que não oferecem o serviço de internet para dispositivos móveis com qualidade os aplicativos podem ser executados mesmo sem a internet.

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

As tecnologias oferecidas pelo iOS tornam o desenvolvimento mais fácil e seguro. Se as tecnologias das camadas superiores não forem suficientes para o processo de desenvolvimento, pode se utilizar das camadas inferiores que são mais flexíveis e permite um maior controle sobre os recursos utilizados no dispositivo.

2.3 DESENVOLVIMENTO NATIVO

Aplicações nativas são aquelas desenvolvidas especificamente para serem executadas em uma determinada plataforma, Android, iOS etc., de um dispositivo móvel que permite que os recursos destes aparelhos sejam explorados ao máximo.

Utilizam os componentes básicos de interface de cada plataforma, aumentam a complexidade e os custos de desenvolvimento já que não é possível a reutilização de código entre as plataformas (SAMBASIVAN, 2011). Uma equipe que desenvolve exclusivamente para Android provavelmente não terá a mesma competência desenvolvendo para iOS ou Windows Phone. O desenvolvimento nativo é normalmente feito quando existem elevados requisitos de desempenho e usabilidade, é necessário usar grande quantidades de animações, como em jogos por exemplo, acessar APIs específicas do dispositivo, como o acelerômetro ou a aplicação é exclusivamente para usuários de uma determinada plataforma. São desenvolvidas usando linguagens de programação de alto nível, como por exemplo, o Java para o Android, Objective-C para iOS ou C# para Windows Phone (WHITE, 2013). As APIs nativas são fornecidas ao desenvolvedor junto com o SDK da plataforma. As APIs das plataformas são normalmente desenvolvidas para fornecer aplicativos nativos de acesso ideal para as capacidades de *hardware*, como a câmera do dispositivo móvel, e emparelhamento de *bluetooth*. Outras funcionalidades incluem a capacidade do uso da aplicação pelo usuário sem conexão com a internet.

A aplicação desenvolvida nativamente geralmente tem uma maior integração visual com a plataforma do dispositivo móvel, isto porque os componentes visuais como botões, menus e listas, por exemplo, são fornecidos e estilizados pela própria plataforma. Nesta arquitetura de desenvolvimento é possível utilizar melhor os recursos de hardware disponíveis no dispositivo móvel, como acelerômetro, bússola, GPS, e câmera, que normalmente não podem ser acessadas diretamente por um aplicativo web executado em um navegador de internet. Isto possibilita a criação de

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

uma aplicação muito mais bem elaborada em termos de funcionalidades e integração com o dispositivo móvel.

Normalmente em uma aplicação desenvolvida nativamente é feita uma validação e publicação em uma loja de aplicações no qual os usuários poderão instalá-la em seus *smartphones* e posteriormente acessá-la através de um ícone no menu de aplicações ou na tela inicial do aparelho. Dessa forma o fabricante da plataforma é responsável por toda a infraestrutura para a publicação e distribuição dos aplicativos, assim, quem desenvolve a aplicação pode focar no desenvolvimento do aplicativo em si, não tendo preocupações por qual forma o usuário irá adquirir a aplicação.

2.4 DESENVOLVIMENTO HÍBRIDO

Os aplicativos híbridos são parcialmente nativos e parcialmente Web Apps. Como os nativos, eles devem ser baixados através de um aplicativo de loja, ficam armazenados na tela principal do dispositivo e podem aproveitar todas as funcionalidades do dispositivo (câmera, GPS, acelerômetro, gestos etc.).

Considerando um cenário com notável diferença entre sistemas operacionais e plataformas de programação existentes, a criação de aplicativos híbridos tem por finalidade funcionar em qualquer tipo de dispositivo e para as diferentes plataformas como iPhone, Android, Windows Phone etc., utilizando o mesmo código fonte.

Segundo Ribeiro e Freire (2013, p13), “definimos um *framework* multiplataforma como um conjunto de arquivos de códigos fonte, bibliotecas e ferramentas que oferece suporte a mais de uma plataforma, pelo menos duas diferentes”. Nesta definição, iOS e Android são duas plataformas diferentes, iOS 5 e iOS 6 não são, e permite o desenvolvimento sem ramificações de código fonte, ou seja, o mesmo código para todas plataformas.

De acordo com Palmieri (2012), estas ferramentas trouxeram os seguintes benefícios: redução da complexidade, redução de código, redução do tempo de desenvolvimento e manutenção, diminuição de conhecimento necessário sobre a API, maior facilidade no desenvolvimento e aumento de participação no mercado.

O desenvolvimento de uma aplicação híbrida resolve o problema usando tecnologias Web baseadas em padrões nativos. As aplicações “híbridas” são uma categoria especial de aplicações Web que ampliam o ambiente do aplicativo

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

baseado na Web com o uso de APIs nativas da plataforma disponível em determinado dispositivo. O padrão de uma aplicação híbrida é desenvolvido para ser executado em ambos os ambientes, móvel e desktop. A aplicação híbrida é executada dentro de um ambiente de processo nativo no dispositivo assim como a aplicação nativa. Essas aplicações envolvem o conteúdo HTML dentro de um controle de navegador em modo de tela cheia sem deixar visível a barra de endereços ou qualquer outro controle do navegador. Aplicações híbridas usam uma camada chamada “camada de ponte”, que permite que o JavaScript acesse muitos recursos específicos do dispositivo e APIs nativas que geralmente não são acessíveis a partir do navegador Web móvel.

Por estar sempre disponível no dispositivo mesmo sem a presença de internet, a aplicação híbrida exige que as bibliotecas fiquem armazenadas localmente no dispositivo.

Uma vez que o aplicativo é multiplataforma ele pode ser empacotado como um aplicativo nativo, e este é desenvolvido uma única vez e poderá ser comercializado nas diversas *app stores.*

Para Hartmann (2011, p6), as ferramentas de desenvolvimento são divididas nos seguintes grupos:

Biblioteca: Pequeno kit de ferramentas que oferece funcionalidades muito específicas ao usuário, normalmente utilizada em conjunto com outras bibliotecas e ferramentas para o desenvolvimento de aplicativos

Framework: conjunto de bibliotecas, componentes de software e diretrizes de arquitetura que fornecem ao desenvolvedor um conjunto de ferramentas abrangente para criar uma aplicação móvel completa.

Plataforma: Um conjunto de *frameworks*, ferramentas e serviços que não apenas permitem ao desenvolvedor construir uma aplicação para dispositivo móvel completa, como também empacotá-las para distribuição nas *app stores* ou na nuvem. Geralmente possuem algum ambiente de desenvolvimento integrado para facilitar a construção, documentação, apoio e ferramentas de automação.

Produto/serviço: Fornece funcionalidade ou serviço pronto para ser usado por um aplicativo em um dispositivo móvel.

Definir as diferentes abordagens utilizadas pelas ferramentas de desenvolvimento de aplicações multiplataforma para dispositivos móveis é

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

fundamental, pois para cada tipo de abordagem há distintas limitações e podem atender diferentes propósitos.

2.5 DESENVOLVIMENTO DE WEBAPPS

Aplicações para *smartphones* desenvolvidas com recursos WEB são como "WebApps" e se destacam por sua simplicidade por intermédio de texto e informações gráficas com poucos recursos (Pressman,2011). Porém, com a popularidade da WEB 2.0, ocorreu um grande avanço dessas aplicações, que não só se limitam a modernos ambientes, mas oferecendo recursos avançados ao usuário final, desde integração com bancos de dados até incorporando sistemas comerciais

Estes aplicativos são sites desenvolvidos com tecnologias voltadas para a web (HTML, CSS e Java Script) e aperfeiçoados para funcionar como um aplicativo. A aplicação é executada em um navegador do dispositivo móvel, o que a torna compatível com todas as plataformas e de fácil atualização. O comportamento em cada plataforma depende da maneira como o navegador processa o aplicativo. Web Apps apresentam algumas desvantagens em comparação com aplicações nativas. O acesso aos recursos do dispositivo é limitado e a aplicação não é instalada no dispositivo e geralmente só funciona com o dispositivo conectado à internet.

O acesso inicial a esses dispositivos é feito como se fosse acessada uma determinada URL e tem a opção de "instala-lo" na tela inicial de seu dispositivo, mas na verdade é criado um atalho para a página onde hospeda o serviço a ser utilizado.

Tem acesso a algumas funcionalidades análogas a uma aplicação nativa como esconder botões do navegador, gestos de navegação, e com o cache do navegador, em algumas aplicações é possível a visualização off-line, usar GPS e link para ligações diretas.

Ao falarmos em web apps, falamos em aplicativos mais leves, que consomem menos memória do dispositivo, já que não são instaladas no mesmo, pois o serviço que será utilizado está hospedado em um link que é acessado pelo ícone de acesso a ele, um exemplo de web app é o Facebook Lite, o aplicativo nativo do Facebook é capaz de consumir muita memória tanto de armazenamento, quanto de processamento de nosso dispositivo, por isso uma alternativa é a do Facebook Lite, um web app que utiliza a memória de cache do navegador que ele é acessado, e

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

seu processamento é executado em seu host, diminuindo assim o consumo em nosso dispositivo, porém, as suas funcionalidades são limitadas perante ao app nativo.

2.6 REACT NATIVE

O React Native é uma ferramenta desenvolvida pela empresa Facebook que permite a possibilidade de criação de aplicativos para dispositivos móveis de forma nativa para Android e iOS, utiliza as ferramentas de front-end mais modernas e o desenvolvimento e é baseado em JavaScript.

O React Native funciona através de uma ponte que é responsável por realizar a comunicação entre a camada JavaScript e a camada nativa da aplicação, é um código JavaScript rodando em uma máquina virtual, controlado por uma interface de usuário nativa, o código gerado por JavaScript não é compilado ou convertido para a linguagem nativa do dispositivo. Tudo acontece através dessa ponte entre o JavaScript e o ambiente nativo do dispositivo (MONTEIRO, 2017).

De acordo com Eisenman (2016, p. 17) “React Native invoca as APIs de renderização nativas em Objective-C, para iOS ou Java para Android”.

A utilização do *framework* tem como vantagens uma experiência do usuário muito mais fluida, os carregamentos e requisições acontecem mais rapidamente, pois não é necessário uma *webview* para intermediar o processo, maior integração entre o hardware do dispositivo, maior nível de segurança entre aplicativos web e uma melhor performance da aplicação.

O desenvolvimento é extremamente mais rápido, possuindo o reaproveitamento do código gigantesco que não existiria se fosse desenvolvido em Objective-C para iOS e Java para Android (KUPKA, 2017).

2.7 FLUTTER

O Flutter é um framework de código aberto desenvolvido pelo Google para desenvolvimento de aplicativos para dispositivos móveis que executem tanto no Android quanto no iOS a partir de um único código, com o objetivo de permitir a criação de aplicativos de alta performance para ambas as plataformas com uma experiência nativa (CORAZZA, 2018).

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

Seu desenvolvimento é direcionado ao design e os *widgets* são os blocos básicos da interface de usuário de um aplicativo desenvolvido com Flutter, assim eles definem elementos estruturais, elementos de estilos e aspectos de layout. Segundo o site Flutter.dev[5] os *widgets* do Flutter incorporam todas as diferenças críticas de plataforma, como rolagem, navegação, ícones e fontes, para fornecer desempenho nativo completo no iOS e no Android.

Além de usar linguagem Dart, uma linguagem de programação moderna, precisa e fortemente tipada e orientada a objetos, que pode ser compilada no modo *Just in time*. O que torna o Flutter diferente da maioria dos frameworks para desenvolvimento de aplicativos móveis, é que ele não utiliza os widgets fornecidos com o dispositivo, ele utiliza seu próprio mecanismo de renderização de alto desempenho para desenhar widgets.

Aplicações com Flutter são mais rápidas em comparação com as demais aplicações multiplataforma, pois por possuir *widgets*, tem uma interface leve, além de ser compilado para o código nativo, aplicativos com Flutter funcionam quase como um aplicativo nativo por isso a maioria dos aplicativos híbridos, perde em desempenho para o Flutter

3 Resultados e discussão

O presente estudo teve base em uma pesquisa bibliográfica, tendo sido examinados artigos acadêmicos e livros de desenvolvimentos de tecnologias sobre assunto. Foram analisados os métodos de desenvolvimento de cada aplicação, bem como suas características e suas particularidades afim de se expor os pontos positivos, negativos e desempenho bem como as linguagens de programação de desenvolvimento de cada aplicação.

Após análise da base de estudos e verificados os diferentes tipos de desenvolvimento concluiu-se que não há um melhor e um pior método de desenvolvimento, mas sim há a necessidade de cada aplicação em relação ao *hardware*, necessidade de trabalho off-line entre outros recursos. Quando uma aplicação necessita do máximo desempenho do hardware como um game por exemplo a melhor forma de desenvolvimento é a nativa já que há uma maior

[5] https://flutter.dev/

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

interação da aplicação com o dispositivo aproveitando o máximo do *hardware* disponível e com a possibilidade de execução off-line já que a aplicação é instalada no aparelho, ou quando a aplicação deverá ser exclusiva de determinada plataforma, assim não poderá ser instalada em dispositivos de plataformas diferentes.

Já em uma aplicação que não necessite de máxima integração com o *hardware* do dispositivo, porém tem a necessidade de execução off-line, e a intenção é atingir um número maior de usuários, como no caso de um navegador GPS por exemplo, a melhor forma de desenvolvimento é a híbrida, já que o dispositivo é instalado no aparelho, poderá ser executado off-line, há a interação da aplicação com o GPS do dispositivo, os custos de desenvolvimento são menores e pode ser instalado em diferentes plataformas.

Mas se uma aplicação necessita de conexão com internet, tem muito pouca ou nenhuma interação com o hardware do dispositivo e não tem necessidade de execução off-line, e tem-se a intenção de atingir o maior número de usuários possíveis, como um app de delivery por exemplo, a melhor forma de desenvolvimento é uma web app, já que é desenvolvida como uma página web seu custo é bem menor, suas atualizações são feitas diretamente na página web, e pode ser executada em qualquer tipo de plataforma já que é apenas criado um atalho para a página onde é hospedada o serviço e tem a vantagem de ocupar apenas a memória cache do navegador, não ocupando espaço na memória do dispositivo

A tabela 1 faz um comparativo entre o percentual de perguntas, o ranking e o percentual de usuários por linguagem de programação.

Tabela 1: Tabela Comparativa por Linguagem de Programação

	NATIVO	HÍBRIDO	WEBAPPS	REACT NATIVE	FLUTTER
Linguagem de programação	Android: Java ou Kotlin iOS: Objective-C ou Swift	Html, Css, JavaScript	Html, Css, JavaScript	JavaScript e JSX	Dart

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

Percentual de perguntas no StackOverflow[6]	Java: 45,3% Kotlin: 4,5% Objective-C:7% Swift: 8,1%	JavaScript: 69,8% HTML: 68,5% CSS: 65,1%	JavaScript: 69,8% HTML: 68,5% CSS: 65,1%	JavaScript: 69,8% JSX: não informado	Dart: não informado
Ranking da linguagem de Programação[7]	Java: 1º 15% Kotlin:35º 0,33% Objective-C:11º 1,5% Swift: 19º 1%	JavaScript: 7º 2,5%	JavaScript: 7º 2,5%	JavaScript: 7º 2,5%	Dart: 26º 0,5%
Exemplo de aplicativos	WhatsApp Google Maps Skype	Twitter Yelp Uber	Facebook Lite Linkedin	Facebook Instagram AirBnB	Alibaba Google Ads AppTree

Um dos critérios para se adotar uma linguagem de programação ou plataforma é analisar o tamanho da comunidade de desenvolvedores ao redor do mundo que já a adotam. Desta forma a Tabela Comparativa apresenta o percentual de perguntas no StackOverflow relacionada à linguagem de programação, e o Ranking desta linguagem segundo o site Tiobe (2019). Observa-se que a linguagem JavaScript é largamente utilizada por desenvolvedores pois lidera o número de perguntas feitas a respeito da linguagem e é a sétima mais utilizada segundo o site, e sendo ela utilizada por três dos métodos de desenvolvimento apresentados.

A linguagem Java também bastante usada é a primeira do ranking também tendo um considerável percentual de perguntas relacionadas á linguagem e é utilizada para o desenvolvimento Android que domina grande parte do mercado.

Já a linguagem Dart utilizada pelo Flutter ainda tem uma posição baixa no ranking, mas vale salientar que ainda é uma linguagem nova e por ser semelhante ao JavaScript tem uma curva de aprendizagem menor e com grande potencial de crescimento entre os desenvolvedores.

Ainda na Tabela Comparativa, são apresentados alguns exemplos de aplicativos conhecidos que utilizam cada uma das plataformas. Essa métrica é válida para verificar a adoção da plataforma pela indústria, além de proporcionar maior

[6] https://insights.stackoverflow.com/survey/2018#technology
[7] https://www.tiobe.com/tiobe-index/

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

segurança de que esta plataforma será mantida, exemplos como Facebook e WhatsApp que são aplicativos com um gigantesco número de usuários, assim como as demais redes sociais, que estão instaladas na grande maioria dos dispositivos móveis, nos dão uma visão de como os métodos de desenvolvimento atendem as diferentes características de cada *app* cabendo ao desenvolvedor uma análise de qual método melhor lhe atenderá.

Considerações finais

Com o crescimento da tecnologia na área de aplicativos móveis, têm surgido grandes desafios para programadores, desenvolver para diferentes plataformas com o mínimo custo possível, com o mínimo de erros, e com máximo desempenho possível, é uma tarefa que exige muito trabalho e dedicação.

Este artigo apresenta resultados de uma pesquisa na área de desenvolvimento de aplicativos móveis expondo algumas formas de desenvolvimento diferentes afim de se analisar qual método é mais eficaz diante das necessidades de um aplicativo.

A pesquisa conclui que não se tem um método de desenvolvimento ideal para qualquer tipo de aplicação, mas sim, um método que atende as particularidades de cada aplicação, sendo possível a análise de cada método e a escolha da melhor forma de desenvolvimento para a aplicação, atendendo aos requisitos necessários, seja ele de performance, conectividade ou até mesmo de custo de desenvolvimento.

Embora o desenvolvimento de web apps ainda tenha suas limitações, com o surgimento de novas tecnologias como o PWA (*Progressive Web Apps*), possa solucionar os problemas de acesso a hardware que o método possui, podendo atender a uma gama maior de requisitos para desenvolvimento de aplicativos móveis.

Como trabalho futuros, pode-se avaliar a performance do aplicativo desenvolvido em cada uma das diferentes tecnologias em termos de velocidade do carregamento do app, tempo de atualização das telas e performance do processamento de dados. Também pode-se realizar estudos para avaliar a curva de aprendizado e produtividade no desenvolvimento de cada uma das tecnologias.

ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

Referências

ANDRADE, Alisson Wilker; AGRA, Ronaldo; MALHEIROS, Viviane. Estudos de caso de aplicativos móveis no governo brasileiro. Serviço Federal de Processamento de Dados, SERPRO, Brasília, 2013.

BASTOS GUIMARÃES, Ítalo José; FERREIRA DE SOUSA, Marckson Roberto. Reflexões sobre Arquitetura da Informação para dispositivos móveis. **Em Questão**, v. 22, n. 1, 2016.

CORAZZA, Paulo Victor. Um aplicativo multiplataforma desenvolvido com flutter e NoSQL para o cálculo da probabilidade de apendicite. 2018.

CRUZ, Vitor Silva; PRETUCELLI, Erick Eduardo. TECNOLOGIAS WEB PARA O DESENVOLVIMENTO MOBILE NATIVO. 2017.

DA SILVA, Marcelo Moro; SANTOS, Marilde Terezinha Prado. Os paradigmas de desenvolvimento de aplicativos para aparelhos celulares. **Revista TIS**, v. 3, n. 2, 2014.

DEVELOPERS, Android. What is android. 2011.

DOS SANTOS MONTAN, Jardel et al. AVALIAÇÃO DE PLATAFORMAS HÍBRIDAS PARA DESENVOLVIMENTO DE APLICAÇÕES PARA O ANDROID-EVALUATION OF HYBRID ANDROID DEVELOPMENT PLATFORMS. **Multiverso: Revista Eletrônica do Campus Juiz de Fora-IF Sudeste MG**, v. 2, n. 2, p. 116-127, 2017.

GOMES, RAFAEL CAVEARI et al. Sistema Operacional Android. Universidade Federal Fluminense, 2012.

PAULINO, Rita de Cássia Romeiro. Conteúdo digital interativo para tablets-iPad: uma forma híbrida de conteúdo digital. **Revista de Estudos da Comunicação**, v. 14, n. 33, 2017.

SABOIA, Juliana; VARGAS, PL de; VIVA, M. A. O uso dos dispositivos móveis no processo de ensino e aprendizagem no meio virtual. Revista Cesuca Virtual: conhecimento sem fronteiras, v. 1, n. 1, p. 1-13, 2013

PELLANDA, Eduardo Campos. Comunicação móvel no contexto brasileiro. **Comunicação e mobilidade**, p. 11, 2009.

PEDRASSANI, Carlos Eduardo. Uma solução em Nodejs e react native para busca e oferta de emprego. 2018.

PREZOTTO, Ezequiel Douglas; BONIATI, Bruno Batista. Estudo de frameworks multiplataforma para desenvolvimento de aplicações mobile híbridas. **Universidade Federal de Santa Maria, Trabalho de Conclusão de Curso**, 2014.

ROCHA, Adriano Mendonça; NETO, Roberto Mendes Finzi. Introdução a Arquitetura Apple iOS. 2014.

 ISSN: 2595-5942

Revista EduFatec: educação, tecnologia e gestão
V.2 N.1 - janeiro-junho/2019

SANTOS, Luís Duarte de Jesus. **Desenvolvimento de soluções mobile para o Banco de Portugal**. 2014. Tese de Doutorado.

.

SILVA, LLB; PIRES, Daniel Facciolo; NETO, Silvio Carvalho. Desenvolvimento de aplicações para dispositivos móveis: tipos e exemplo de aplicação na plataforma IOS. **Franca/SP**, 2015.

TAVARES, Henrique Leal. Introdução a Desenvolvimento de Aplicações Híbridas. **Revista Eletrônica eF@ tec**, v. 6, n. 1, p. 11-11, 2016.

ISSN: 2595-5942