

COMPARAÇÃO DE CUSTO-BENEFÍCIO ENTRE OS SDKS IONIC E FLUTTER E A BIBLIOTECA REACT NATIVE PARA APLICAÇÕES MÓVEIS

Giovanni Jatene Barberato¹; Gabriel Ferreira Silva²; Thiago Silva Ferreira³,
José Roberto de Almeida⁴

^{1, 2, 3, 4} UNIUBE - Universidade de Uberaba

giovanni.barberato@edu.uniube.br, jose.almeida@uniube.br

Resumo

Com o uso cada vez maior dos *smartphones*, foram surgindo aplicativos que atendessem às necessidades dos usuários. Um problema atual é o desenvolvimento de aplicativos para sistemas diversos, o que gera um alto custo de capacitação para os desenvolvedores, além de tempo para adequar aos requisitos de cada sistema. Para facilitar esse processo de migração entre as plataformas, foram desenvolvidas linguagens híbridas que permitem que um mesmo *software* seja escrito para sistemas diferentes. Cada linguagem e suas bibliotecas possuem vantagens e desvantagens que precisam ser analisadas para uma escolha adequada, visando uma menor curva de aprendizagem e um menor custo para implantação. Para auxiliar nesse processo, foram desenvolvidos *softwares* simples, calculadoras para avaliar a análise dos *frameworks* Ionic, React Native e do SDK Flutter. Também foi realizada uma avaliação com base nos conhecimentos e análises dos autores estes desenvolviam os programas. O resultado alcançado aborda estudos e discussões dos autores sobre o tema proposto buscando a melhor resposta para o processo de comparação das plataformas, levando em conta as atuais versões de cada linguagem, além de estabilidade de *software*, adaptação para múltiplos sistemas operacionais, futuras melhorias para ampliação dos recursos das calculadoras desenvolvidas e recursos necessários para instalação nos computadores de desenvolvimento.

Palavras-chave: Internet. HTML. Javascript. Multiplataforma.

1 Introdução

Este artigo tem como objetivo apresentar uma comparação entre os *frameworks* Ionic, React Native e Flutter devido às dificuldades encontradas no desenvolvimento móvel utilizando as linguagens nativas de cada tecnologia, que tornam o desenvolvimento de aplicativos lento e complexo, tornando-os de alto custo, devido ao seu alto nível de complexidade e elevada curva de aprendizagem. Para isso foram desenvolvidas linguagens híbridas, que embora facilitem o desenvolvimento para múltiplas plataformas, a escolha da mais vantajosa se torna difícil por causa da imensa variedade de linguagens e suas diversas complexidades.

Com o aumento do uso dos *smartphones* e equipamentos IOT (*Internet of things*, em português, internet das coisas) é necessário que se utilize uma linguagem de programação adaptável às constantes evoluções tecnológicas e que permita que os desenvolvedores criem de forma fácil e simples as suas aplicações.

Segundo Avelino (2017), uma fase inicial utilizada para o desenvolvimento de aplicativos é a escolha das linguagens mais adequadas e com maior benefício para os desenvolvedores, já que diferentes sistemas operacionais favorecem diferentes linguagens.

A escolha do *framework* deve ser feita de acordo com as necessidades e critérios de desenvolvimento do produto, sendo alguns fatores que influenciam: popularidade, tamanho da comunidade, sustentabilidade, suporte, técnicas de programação, segurança, documentação,

licenciamento pago ou gratuito e a disponibilidade de recursos no mercado.

2 Materiais e Métodos

Foram realizadas pesquisas em artigos e análises sobre linguagens híbridas de programação utilizando a internet. Com esses dados, foram feitos estudos sobre SDK, ou *Kit* de desenvolvimento de software, Flutter, *frameworks* Ionic e biblioteca React.

2.1.1 Flutter

O Google desenvolveu um SDK chamado Flutter que é baseado na linguagem Dart. Segundo Coraza (2018), essa plataforma possui código aberto e tem capacidade para o desenvolvimento de aplicativos nos sistemas Android e IOS, utilizando um mesmo código. Esse recurso permite aos programadores a criação de *softwares* de alto desempenho, similar aos desenvolvidos em linguagens nativas de ambos os sistemas operacionais.

De acordo Oliveira e Oliveira (2019), a facilidade oferecida pelo Flutter, permite que os desenvolvedores compartilhem os pacotes criados, o que amplia o número de componentes disponíveis para desenvolvimento.

2.1.2 Ionic

Para Passos (2018), o *framework* conhecido como Ionic, utilizado para o desenvolvimento de aplicativos para dispositivos móveis, utiliza recursos como: preenchimento de conteúdo através de HTML, o desenvolvimento de estilos é feito pelo Sass, similar ao CSS, além de utilizar Javascript e TypeScript presente no Angular para codificação das funções. Como é um *framework front-end*, possui a vantagem de ser utilizado por *navegadores de internet* para o processo de desenvolvimento de *software*. Com isso, é

possível utilizar o recurso de *debug* dos navegadores para identificação de inconsistências que causem erros JANONES (2016).

Para Huynh (2017), a vantagem desse método é criar um aplicativo para dispositivos móveis que funcione baseado em tecnologia *web*.

2.2.2 React Native

Outra plataforma utilizada para o desenvolvimento de aplicativos é o React Native. Segundo Becker (2019) ele é um *framework* baseado no React, desenvolvido pela equipe do Facebook, que permite a programação de aplicativos para dispositivos móveis que utilizem os sistemas Android e IOS, usando como base de código o Javascript. No término do desenvolvimento, a aplicação é convertida para a linguagem nativa de cada sistema operacional.

Para Silva (2019), o React também é uma biblioteca JavaScript conhecida por suas facilidades na construção de interfaces na internet. Sua principal característica é a capacidade de simplificar ambientes complexos que precisam de tratativas de eventos e manipulação de dados.

De acordo com Salohonka (2020), uma característica do React Native é o uso da componentização de objetos simples como textos e imagens para a formação de coleções de componentes.

Após os estudos, foram desenvolvidos *softwares* de calculadora simples como a do Windows.

Para avaliar as características de cada SDK, foi elaborado um questionário para pontuação pelos desenvolvedores do artigo conforme a seguir:

2.1 Questões da Avaliação

- 1) O desenvolvedor tem os conhecimentos necessários para utilizar a biblioteca ou *framework*?

- 1- Não, possui apenas conhecimento em lógica básica;
 - 2- Pouco, possui conhecimento em lógica básica e algum conhecimento em programação.
 - 3- Razoável, possui conhecimento em lógica básica, programação e algum conhecimento em desenvolvimento *web*.
 - 4- Intermediário, possui conhecimento em lógica básica, programação, conhecimento em desenvolvimento *web* e algum conhecimento em aplicativos móveis.
 - 5- Avançado, o desenvolvedor já faz uso das plataformas biblioteca ou *framework*.
- 2) A plataforma possui bom desempenho para o desenvolvimento de aplicações móveis e *web*?
- 1- Não, é lenta e não atende às necessidades das aplicações na maioria das vezes.
 - 2- Desempenho baixo, atende em parte as necessidades das aplicações, mas para gerar aplicações é lenta.
 - 3- Razoável, atende algumas das necessidades das aplicações e possui um bom desempenho.
 - 4- Muito bom, atende a maioria das necessidades das aplicações e tem um desempenho rápido.
 - 5- Excelente, atende todas as necessidades das aplicações e tem um desempenho muito rápido.
- 3) Qual o grau de dificuldade de aprendizado do *framework* ou biblioteca?
- 1- Fácil, em poucos dias é possível aprender a utilizar e não é necessário experiência em desenvolvimento *web* ou *mobile*.
 - 2- Pouco fácil, é necessária alguma experiência no desenvolvimento *web* ou *mobile* para aprender.
 - 3- Intermediário, é necessária experiência no desenvolvimento *web* ou *mobile* para aprender.
 - 4- Difícil, é preciso de experiência avançada em de desenvolvimento *web* ou *mobile* para aprender.
 - 5- Muito Difícil, é preciso de experiência avançada em de desenvolvimento *web* ou *mobile* para aprender e além de outros conhecimentos intermediário ou avançado em programação orientada a objeto.
- 4) Qual a capacidade do *framework* ou biblioteca de desenvolver aplicações híbridas para dispositivos Android e IOS?
- 1- Muito Baixa, o *framework* ou biblioteca permite desenvolver melhor para uma plataforma específica.
 - 2- Baixa, o *framework* ou biblioteca permite desenvolver melhor para uma plataforma específica, demais plataformas funcionam com dificuldade.
 - 3- Razoável, o *framework* ou biblioteca permite desenvolver para ambas as plataformas, mas as aplicações não são otimizadas.
 - 4- Alta, o *framework* ou biblioteca permite desenvolver para ambas as plataformas e as aplicações são otimizadas.
 - 5- Muito Alta, o *framework* ou biblioteca permite desenvolver para ambas as plataformas e as aplicações têm o mesmo desempenho de aplicações nativas.

3 Resultados obtidos

Após o desenvolvimento dos softwares e das avaliações individuais dos desenvolvedores, foi feito um quadro com as notas finais obtidas, além de justificativas para os resultados apresentados

Notas: 1 a 5.

Quadro I - Tabela de Avaliação

Questionário	Flutter	Ionic	React
1- O desenvolvedor tem os conhecimentos necessário para utilizar a biblioteca ou framework?	5	3	4
2- A plataforma possui em bom desempenho para o desenvolvimento de aplicações móveis e web?	4	5	4
3- Qual o grau de dificuldade de aprendizado do framework ou biblioteca?	3	4	2
4- Qual a capacidade do framework ou biblioteca de desenvolver aplicações híbridas para dispositivos Android e IOS?	5	5	4
Média	4,25	4,25	3,5

Fonte: Elaborado pelos autores.

3.1 Justificativa dos resultados

Questão 1 - O que influenciou as notas, foi a experiência pessoal de cada um, por isso quem já trabalhou com desenvolvimento avaliou com notas mais altas.

Questão 2 – Embora todas as plataformas possuam um desempenho muito bom, quem levou vantagem nesse caso foi o Ionic, por sua familiaridade com o padrão *HTML 5* e teve um desempenho melhor para *web*.

Questão 3 – Aprender novas tecnologia é um processo demorado e quem levou vantagem foi o Ionic, novamente pela sua familiaridade com o padrão *HTML 5*, o que facilita o aprendizado.

Questão 4 – Aqui houve um empate entre Flutter e Ionic, devido a forma clara que expõe as aplicações Android e IOS, facilitando para o desenvolvedor acompanhar o processo de criação para cada plataforma.

4 Discussão

Diante do exposto, foi considerado que em primeiro lugar fica o Ionic, devido a sua facilidade para desenvolvimento que lembra o *HTML 5*, além de oferecer suporte para desenvolvimento *Web* e *mobile*. Em segundo lugar o Flutter em razão de sua facilidade de usar bibliotecas nativas do Dart e por último o React que utilizou o Expo que é uma ferramenta comum no Ract native. para facilitar a utilização das APIs nativas, diferentes das demais linguagens que não precisaram.

5 Conclusão

Conclui-se que no desenvolvimento de *software* realizado, que o Ionic apresenta melhores adaptações para desenvolvimento tanto *web* quanto *mobile*. Embora o Flutter apresente capacidade para desenvolvimento *Web* e para aplicativos nativos em Windows e Mac OS, o Ionic continua possuindo vantagem devido a sua estabilidade, mas futuramente com a evolução da plataforma da Google, o Flutter será mais vantajoso.

Referências

AVELINO, André. **Quais são as melhores linguagens de programação para app developers?** [s. l.]: Eu na TI, 2017.

Disponível em:

<https://www.eunati.com.br/2017/10/quais-sao-as-melhores-linguagens-de-programacao-para-app-developers.html>. Acesso em: fev. 2020.

BECKER, Lauro. **O que é React Native?** [s. l.]: Orgânica Digital, 2019. Disponível em: <https://www.organicadigital.com/blog/o-que-e-react-native/>. Acesso em: set. 2020.

CORAZZA, Paulo Victor. **Um aplicativo multiplataforma desenvolvido com Flutter e NoSQL para o cálculo da probabilidade de apendicite**. Orientador: Leandro Krug Wives e Ricardo Francalacci Savaris. 2018. 64 f. Monografia (Bacharelado em Ciência da

Computação) – Instituto de informática, Universidade do Rio Grande do Sul, Porto Alegre, 2018.

HUYNH, Minh. **Hybrid app approach: could it mark the end of native app domination?** Informing Science Institute. Volume 14, 2017. Disponível em: <https://core.ac.uk/reader/206258250> Acesso em: nov. 2020.

JANONES, Ramos de Souza. **Aplicativos mobile com o Angular e Ionic.** [s. l.]: Ramos da informática, 2016. Disponível em: <https://www.ramosdainformatica.com.br/mundojs/ionic-framework/aplicativos-mobile-com-o-angularjs-e-ionic-framework/>. Acesso em: set. 2020.

MADUREIRA, Daniel. **Aplicativo nativo, web app ou aplicativo híbrido?** [s. l.]: Usemobile, 2017. Disponível em: <https://usemobile.com.br/aplicativo-nativo-web-hibrido/>. Acesso em: set. 2020.

OLIVEIRA, Filipe; OLIVEIRA, Elthon. **Aprendizado de máquina aplicado a dados agrometeorológicos.** In: ESCOLA REGIONAL DE COMPUTAÇÃO BAHIA, ALAGOAS E SERGIPE (ERBASE) , 2019,

Ilhéus. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019 . p. 101-106. Disponível em: <https://sol.sbc.org.br/index.php/erbase/article/view/8962/8863> Acesso em: nov. 2020.

PASSOS, Clayton K. N. **O que é o IONIC ?** [s. l.]: Tableless, 2018. Disponível em: <https://tableless.github.io/iniciantes/manual/js/o-que-framework.html>. Acesso em: set. 2020.

SALOHONKA, Matias. **Automated testing of react native applications.** Orientador: Professor Ari Happonen. 2020. 93 p. Tese (Mestrado em Engenharia de Software) - LAPPEENRANTA-LAHTI UNIVERSITY OF TECHNOLOGY LUT, [S. l.], 2020. Disponível em: https://lutpub.lut.fi/bitstream/handle/10024/160734/diplomity%c3%b6_salohonka_matias.pdf?sequence=1&isAllowed=y. Acesso em: nov. 2020.

SILVA, Priscylla. **O que é React e para que serve? | Conversando com o CTO.** [s. l.]: Gobacklog, 2019. Disponível em: <https://gobacklog.com/blog/o-que-e-react-e-para-que-serve/>. Acesso em: set. 2020.