UNIVERSIDADE FEEVALE

LUIZ EDUARDO SANDOVAL VIEIRALVES

##### Análise de linguagem multiplataforma com foco em Flutter

###### Anteprojeto de Trabalho de Conclusão

Novo Hamburgo

2020

LUIZ EDUARDO SANDOVAL VIEIRALVES

##### Análise de linguagem multiplataforma com foco em Flutter

Anteprojeto de Trabalho de Conclusão de Curso, apresentado como requisito parcial

à obtenção do grau de Bacharel em

Ciência da Computação pela

Universidade Feevale

Orientador: Paulo Ricardo Muniz Barros

Novo Hamburgo

2020

RESUMO

Com o crescimento da quantidade de sistemas operacionais, assim como a crescente variação de versões deles, é possível perceber um aumento na dificuldade de manutenção de programas existentes, assim como na criação de novos programas. Essa dificuldade se apresenta tanto para sistemas operacionais diferenciados quanto para versões diferentes de um mesmo sistema operacional. Considerando essa dificuldade, muitas maneiras têm sido apresentadas para realizar o desenvolvimento multiplataforma. *Frameworks* diversificados tem sido apresentados, com propostas de adaptação de linguagens existentes, e até linguagens completamente novas. Com tantas maneiras de desenvolver aplicativos multiplataforma, há a necessidade de avaliar os muitos *frameworks* existentes e realizar comparação deles com o desenvolvimento nativo. Muitos estudos tem focado somente em desenvolvimento *mobile*, havendo no entanto mais ambientes a serem considerados e avaliados como equipamentos ligados ao conceito de IoT(*Internet of Things*) e até computadores tradicionais. Deste modo a proposta aqui apresentada tem como objetivo a avaliação do desenvolvimento multiplataforma, usando a linguagem *Flutter* frente o desenvolvimento nativo, fazendo uso de um *framework* de avaliação que contemple ambientes além do mobile, atendendo as tendências de mercado.

Palavras-chave: Desenvolvimento Multiplataforma; *Flutter*; Avaliação.

SUMÁRIO

MOTIVAÇÃO 1

OBJETIVOS......................................................................................................................4

METODOLOGIA..............................................................................................................5

CRONOGRAMA..............................................................................................................6

MOTIVAÇÃO

Com a democratização da tecnologia, percebe-se um aumento nos ambientes em que um mesmo programa pode ser utilizado (AHMAD et al., 2018), exemplo disso são as redes sociais, que podem ser utilizadas em diferentes ambientes como celulares e computadores. Os desenvolvedores têm buscado essa disponibilidade de manutenção de código simplificada entre muitos ambientes (AHMAD et al., 2018). No entanto pode-se verificar uma grande diferença nos 5 maiores sistemas operacionais, que são os principais fatores nas diferenças de ambiente, e isso somente considerando ambientes nativos mobile e de computadores (STATCOUNTER, 2019), e dentro destes tem-se ainda diferenças grandes de versão (DOBIE, 2012).

Como é possível verificar no gráfico abaixo, tem-se como principais OS utilizados na atualidade (STATCOUNTER, 2019)Android 38. 9%, Windows 35.29%, iOS 14.97%, OS X 8.07%, Linux 0.89%, Desconhecido 0.83%.



Nota-se que esses ambientes têm mantido uma distribuição equivalente no último ano, e que ainda sim existem diferenças de versão que muito afetam a maioria dos ambientes apresentados.

Esse meio distribuído de ambientes tende a aumentar, recentemente a Huawei anunciou seu ambiente multiplataforma o *HarmonyOS*, ambiente este de sistema operacional para diferentes dispositivos (PORTY, 2019), sendo possível o mesmo ambiente ser encontrado em um celular, carro, *wearable* e muitos outros equipamentos. Da mesma maneira o Google busca com o *Fuchsia*, seu sistema operacional multiplataforma , tendo como objetivo apresentado a sua compatibilidade, de maneira que o ambiente possa ser compartilhado, do mesmo modo que tanto uma geladeira quanto um carro possam utilizá-lo, já que precisam de sistemas internos, que além de controle, possam ter uma interface de utilização (ALTAVILLA, 2019).

Como os equipamentos basicamente estão se diversificando e há cada vez mais ligação entre eles tende-se a necessidade de trabalhar na criação de aplicativos que funcionem de maneira clara e em todos eles (MOHAMED; ABDELMOUNAÏM, 2017). Esse processo de programar em vários ambientes traz consigo algumas dificuldades.

Alguns autores destacam como dificuldade o custo necessário para o desenvolvimento de uma aplicação mobile nativa, já que os custos para desenvolver somente para um ambiente já podem ser altos. (KLUBNIKIN, 2019)(DHILLON; MAHMOUD, 2015). Quando o aplicativo deve atender a vários ambientes é necessário replicar o tempo para o desenvolvimento de cada aplicativo, por plataforma, podendo ainda o valor para cada ambiente ser diferenciado (SOMMER; KRUSCHE, 2013), o que gera um crescimento exponencial no custo de desenvolvimento.

Outro aspecto apresentado como dificuldade no desenvolvimento são os diferentes focos das aplicações. Muitas vezes os objetivos a serem alcançados por cada grupo de desenvolvimento podem ser completamente diferentes, analisando por exemplo um aplicativo mobile, que normalmente foca em experiência de usuário e em outras situações em rentabilidade, enquanto um ambiente voltado para carros, precisaria mais foco em segurança (RIEGER; MAJCHRZAK, 2019)

Do mesmo modo há outras características que precisam ser consideradas, como a diferença entre os ambientes de desenvolvimento, a metodologia de testes, a avaliação de usuários, o suporte e validação, entre outros, que dificultam a realização de um projeto estável e contínuo em um ambiente de tecnologia.

Múltiplos ambientes como demonstrado acima adicionam complexidade a um projeto de desenvolvimento, no entanto, pelo ponto de vista dos usuários a facilidade de uso e fluidez, que são essenciais para uma aplicação (MOHAMED; ABDELMOUNAÏM, 2017), para uma escolha baseada em fatos é necessário analisar as ferramentas que são feitas para ambientes multiplataforma, focados em reutilização de código, se possível completa.

Uma das linguagens de programação que vem se mostrando uma possibilidade de resposta a esse cenário é o *Flutter*. *Flutter* é um *framework* multiplataforma, cujo foco inicial seria entregar a capacidade de que, com o mínimo de alteração de código e com a maior velocidade e fluidez na utilização final da aplicação, ainda seja possível desenvolver uma mesma aplicação para vários ambientes. Sendo apresentada pela primeira vez em 2015, e com sua primeira versão estável em 2019 (BOTELLA; ESCRIBANO; PEÑALVER, 2016). O *Flutter* utiliza a linguagem *DART* que apresenta semelhanças a desenvolvimento de linguagens conhecidas, como C e Java, e ainda apresenta e documentação ampla. Este *framework* tem se apresentado como o mais fluido e preparado modelo de desenvolvimento multiplataforma (GOOGLE, 2013; JIM, SIMON et al., 2017).

Essa capacidade tem sido demonstrada por muitas linguagens no meio atual (BIØRN-HANSEN et al., 2019), e por mais que muitas vem sendo utilizadas, o foco tem sido em ambientes mobile, assim como o foco das análises feitas até o momento. Vendo esta situação se faz necessário avaliar linguagens em que o modelo de programação e o suporte a ambientes sejam multiplataforma por completo (BIØRN-HANSEN et al., 2019; RIEGER; MAJCHRZAK, 2019). Para esse meio de análise precisa-se reconhecer que não somente a necessidade do desenvolvedor precisa ser levada em conta, que já considera utilização de recursos nativos e escalabilidade (MOHAMED; ABDELMOUNAÏM, 2017), mas também segurança, estabilidade e performance (RIEGER; MAJCHRZAK, 2019)(DHILLON; MAHMOUD, 2015).

Para que uma análise completa e efetiva seja realizada, é necessário utilizar um modelo de análise para desenvolvimento multiplataforma conforme apresentado por Christoph Rieger e Tim A. Majchrzak(RIEGER; MAJCHRZAK, 2019) no qual deve considerar:

- Necessidades de um desenvolvedor para diminuição de codificação, ainda sim considerando a facilidade e familiaridade do desenvolvedor com a linguagem.

- Capacidade de utilizar, da maneira mais simples possível, funções nativas da plataforma.

- Considerar outras plataformas e não somente mobile no desenvolvimento. Hoje por padrões modelos utilizam somente as plataformas mobile como base para a análise.

- Experiência do usuário na utilização do app desenvolvido pela plataforma.

Para tal análise tende-se a necessidade de levar em conta múltiplos fatores que acarretam pesos diferentes para cada necessidade, podendo ser analisado ambientes de multiplataforma entre si, e com a programação nativa (BIØRN-HANSEN et al., 2019)(AHMAD et al., 2018).

Pode-se assim verificar que é possível utilizar *frameworks* de desenvolvimento multiplataforma para solucionar alguns dos problemas atuais, porém cada solução necessita de componentes e comportamentos específicos, o que leva a necessidade de avaliar qual linguagem ou *framework* é mais adequado. A escolha necessita ser alicerçada em análises sólidas e concisas. Sendo assim este estudo traz como proposta analisar o *Flutter* como *framework* para desenvolvimento multiplataforma em sistemas operacionais como: Android; IOs; Windows e Linux, e compara-lo ao desenvolvimento nativo em relação a usabilidade, capacidade de uso de funções nativas, velocidade de desenvolvimento e possível redução de custo de desenvolvimento .

OBJETIVOS

Objetivo geral

Analisar o *framework* *Flutter*, e sua aplicação para desenvolvimento em múltiplas plataformas não somente dispositivos, mas também plataformas desktop, com seus diversos sistemas operacionais, frente ao desenvolvimento nativo.

Objetivos específicos

* Apresentar um *framework* de avaliação multiplataforma, que não analise somente ambientes de celulares.
* Apresentar as principais características do *Flutter*.
* Desenvolver um modelo de análise específico para avaliação de *Flutter*.
* Desenvolver uma ou mais aplicações para análise com modelos nativos.
* Desenvolver protótipo para avaliação do modelo proposto
* Desenvolver uma pesquisa de avaliação com desenvolvedores nativos, e utilizadores de *Flutter*.
* Avaliar desempenho do *framework* *Flutter*, e verificar sua aplicação aos muitos ambientes atuais.

 METODOLOGIA

A pesquisa em questão tem por objetivo analisar o *framework* *Flutter*, e sua possível aplicação para desenvolvimento em plataformas diversas, comparando com desenvolvimento nativo, fazendo uso de ferramentas e métricas para avaliação e assim o comprar com o desenvolvimento nativo. Para isso será realizado uma pesquisa em *framework*s que possuam ferramentas, artefatos e métricas necessárias para avaliação de desenvolvimento multiplataforma. Assim caracterizando uma pesquisa aplicada, cuja finalidade segundo (PROVDANOV; FREITAS, 2013) objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos.

Para a elaboração do trabalho serão adotados como procedimentos técnicos a pesquisa bibliográfica e a pesquisa experimental. A pesquisa bibliográfica será utilizada pois o trabalho fará o uso materiais publicados em livros, periódicos, teses, dissertações como as apresentadas na IEEE (Institute of Electrical and Electronic Engineers). Já a pesquisa experimental será utilizada por meio de desenvolvimento de aplicação, tanto por meio de desenvolvimento multiplataforma, quanto por meio nativo.

A avaliação será realizada de acordo com o *framework* que está também sendo pesquisado e deverá avaliar de maneira qualitativa, comtemplando do mesmo modo as necessidades de usuários e desenvolvedores. Assim as métricas devem atender às necessidades apontadas por (RIEGER; MAJCHRZAK, 2019), no quesito de diminuição na complexidade de manutenção de aplicativos multiplataforma. Ainda devem atender ao cada vez mais exigente público, não somente mobile, mas multiplataforma por completo. Assim validando o uso do *framework* de desenvolvimento em *Flutter* para multiplataforma em relação a desenvolvimento nativo.

CRONOGRAMA

Trabalho de Conclusão I

|  |  |
| --- | --- |
| Etapa  | Meses |
| Mar  | Abr  | Mai  | Jun  |
| Escrita anteprojeto |  |  |  |  |
| Entrega do anteprojeto |  |  |  |  |
| Realizar pesquisa bibliográfica sobre métodos de avaliaçãode *framework* multiplataforma |  |  |  |  |
| Desenvolvimento de aplicação 1 em *Flutter* |  |  |  |  |
| Desenvolvimento de aplicação 1 em Nativo |  |  |  |  |
| Realizar pesquisa bibliográfica sobre *framework* *Flutter* |  |  |  |  |
| Realizar pesquisa bibliográfica sobre *Flutter* versus Desenvolvimento nativo |  |  |  |  |
| Escrita TC I |  |  |  |  |
| Entrega TC I |  |  |  |  |

|  |  |
| --- | --- |
| Etapa  | Meses |
| Ago | Set | Out | Nov |
| Desenvolver aplicação 2,3 em *Flutter* |  |  | ‘ |  |
| Desenvolver aplicação 2,3 em Nativo |  |  |  |  |
| Desenvolver aplicação 4 em *Flutter* |  |  |  |  |
| Desenvolver aplicação 4 em Nativo |  |  |  |  |
| Aplicar *framework* para análise, e colocar resultados |  |  |  |  |
| Analisar resultados baseados no *framework* escolhido |  |  |  |  |
| Escrita TC2 |  |  |  |  |
| Entrega TC2 |  |  |  |  |
| Defesa TC2 |  |  |  |  |

BIBLIOGRAFIA

AHMAD, A. et al. An Empirical Study of Investigating Mobile Applications Development Challenges. **IEEE Access**, v. 6, p. 17711–17728, 2018.

ALTAVILLA, D. **Google’s Mysterious Fuchsia OS Developer Site Debuts With New Fascinating Details**. Disponível em: <https://www.forbes.com/sites/davealtavilla/2019/06/30/googles-mysterious-fuchsia-os-developer-site-debuts-with-new-fascinating-details/#1dec0af2777f>. Acesso em: 5 abr. 2020.

BIØRN-HANSEN, A. et al. An Empirical Study of Cross-Platform Mobile Development in Industry. **Wireless Communications and Mobile Computing**, v. 2019, 2019.

BOTELLA, F.; ESCRIBANO, P.; PEÑALVER, A. **Selecting the best mobile framework for developing web and hybrid mobile apps**. ACM International Conference Proceeding Series. **Anais**...Association for Computing Machinery, 13 set. 2016. Acesso em: 5 abr. 2020

DHILLON, S.; MAHMOUD, Q. H. An evaluation framework for cross-platform mobile application development tools. **Software - Practice and Experience**, v. 45, n. 10, p. 1331–1357, 1 out. 2015.

DOBIE, A. **Why you’ll never have the latest version of Android**. Disponível em: <http://www.androidcentral.com/why-you-ll-never-have-latest-version-android>. Acesso em: 5 abr. 2020.

GOOGLE. **dart programming language**. [s.l: s.n.]. Disponível em: <https://dart.dev/>. Acesso em: 5 abr. 2020.

JIM, SIMON, G. I. et al. **Flutter Documentation - Flutter**, 2017. Disponível em: <https://flutter.dev/docs>. Acesso em: 5 abr. 2020

KLUBNIKIN, A. **I Want to Create an App. Where Should I Start?** Disponível em: <https://r-stylelab.com/company/blog/mobile-technologies/i-want-to-create-an-app-where-should-i-start>. Acesso em: 5 abr. 2020.

MOHAMED, L.; ABDELMOUNAÏM, A. Decision Framework for Mobile Development Methods. **International Journal of Advanced Computer Science and Applications**, v. 8, n. 2, 2017.

PORTY, J. **Huawei’s new operating system is called HarmonyOS - The Verge**. Disponível em: <https://www.theverge.com/2019/8/9/20798251/huawei-harmonyos-hongmengos-smartphones-internet-of-things-operating-system-android>. Acesso em: 5 abr. 2020.

PROVDANOV, C. C.; FREITAS, E. C. DE. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. [s.l: s.n.].

RIEGER, C.; MAJCHRZAK, T. A. Towards the definitive evaluation framework for cross-platform app development approaches. **Journal of Systems and Software**, v. 153, p. 175–199, 2019.

SOMMER, A.; KRUSCHE, S. **Evaluation of cross-platform frameworks for mobile applications**. Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI). **Anais**...2013Disponível em: <https://dl.gi.de/handle/20.500.12116/17386>. Acesso em: 5 abr. 2020

STATCOUNTER. **StatCounter Global Stats - Browser, OS, Search Engine including Mobile Usage Share**. Disponível em: <https://gs.statcounter.com/>. Acesso em: 5 abr. 2020.