

**UNIVERSIDADE EVANGÉLICA DE GOIÁS
BACHARELADO EM ENGENHARIA DE SOFTWARE**

AS VANTAGENS E DESVANTAGENS DO DESENVOLVIMENTO MULTIPLATAFORMA

**ATIRSON FABIANO BARBOSA DE OLIVEIRA
IAGO GONÇALVES DE ASSIS**

**ANÁPOLIS - GO
2021**

ATIRSON FABIANO BARBOSA DE OLIVEIRA
IAGO GONÇALVES DE ASSIS

AS VANTAGENS E DESVANTAGENS DO DESENVOLVIMENTO MULTIPLATAFORMA

Trabalho de Conclusão de Curso I apresentado como requisito parcial para a conclusão da disciplina de Trabalho de Conclusão de Curso I do curso de Bacharelado em Engenharia de Software da Universidade Evangélica de Goiás

Orientador(a): Prof. M.e William Pereira dos Santos Júnior

Anápolis
2021

ATIRSON FABIANO BARBOSA DE OLIVEIRA
IAGO GONÇALVES DE ASSIS

AS VANTAGENS E DESVANTAGENS DO DESENVOLVIMENTO MULTIPLATAFORMA

Trabalho de Conclusão de Curso I apresentado
como requisito parcial para a obtenção de grau do
curso de Bacharelado em Engenharia de Software
da Universidade Evangélica de Goiás

Aprovado(a) pela banca examinadora em 09 de Junho de 2021, composta por:

Prof. M.e William Pereira dos Santos Júnior
Orientador

Prof. Marcelo de Castro Cardoso

Resumo

Com um aumento da demanda por desenvolvimento de aplicações para dispositivos móveis, foi impulsionado no ano de 2020 mediante a um cenário de pandemia no qual os usuários necessitam de software que ajudem a enfrentar tal adversidade. Se faz necessário o uso de frameworks que atendam às necessidades dos desenvolvedores para que realizem um bom trabalho. O objetivo deste estudo é demonstrar as vantagens e desvantagens do desenvolvimento multiplataforma usando React Native quando comparado com o Java/Kotlin para plataforma nativa Android. E assim evidenciar os custos, mão-de-obra, oportunidades de trabalho, qualidade das ferramentas para desenvolvimento, tempo de esforço empregado na construção de funcionalidades deixando tais informações de forma clara e objetiva. Portanto dando a possibilidade de auxiliar empresas e profissionais que em algum momento irão precisar fazer uma análise de mercado e selecionar um framework que o atenda às suas necessidades e usando critérios objetivos e com base em dados para usar ou descartar tais ferramentas.

LISTA DE ILUSTRAÇÕES

GRÁFICO 1 – Linguagens de programação, script e marcação	18
GRÁFICO 2 – Linguagens e salários anuais	19
GRÁFICO 3 – Parcela das remessas globais de smartphones por sistema operacional de – 2014 a 2023	20
GRÁFICO 4 – Frameworks multiplataforma usado por desenvolvedores de software no – mundo em 2019 e 2020.	20

LISTA DE SIGLAS E ABREVIATURAS

APIS	Application Programming Interface
IDE	Ambiente de Desenvolvimento Integrado
GQM	Goal/Question/Metric

SUMÁRIO

PROBLEMA	8
OBJETIVOS.....	9
OBJETIVO GERAL.....	9
OBJETIVOS ESPECÍFICOS.....	9
JUSTIFICATIVA	10
FUNDAMENTAÇÃO TEÓRICA.....	11
METODOLOGIA.....	13
CRONOGRAMA.....	14
RESULTADOS ALCANÇADOS.....	16
RESULTADOS ESPERADOS	19
REFERÊNCIAS BIBLIOGRÁFICAS.....	20

Problema

O desenvolvimento de software é uma proposta arriscada e cara. Conversar com desenvolvedores de software ilustra rapidamente a situação complicada que o desenvolvimento pode se tornar. Entregas atrasadas, com possíveis defeitos e alterações no produto final, orçamentos, custos e funcionários e partes interessadas frustradas são alguns dos detritos resultantes de uma falha ou menos do que bem-sucedida situação do planejamento de um projeto (ADOLPH; KRUCHTEN; HALL, 2012)

Observando este cenário, enfrentar essa situação de investimento no Brasil, acaba se tornando financeiramente complicada, para uma clareza de valores, de acordo com Bartels (2006, p 1269, apud, ADOLPH; KRUCHTEN; HALL, 2012), “O software à nível global em seu desenvolvimento é uma indústria de 1,6 trilhão de dólares”. Ou seja, é uma parte da indústria que movimento um capital considerável no mercado e que vem ganhando mais espaço a cada dia. Contudo, é válido ressaltar a necessidade de minimizar os custos do desenvolvimento segundo Singh, Singh e Mishra (2018) “Avaliar o custo estimado do projeto, duração e custo de manutenção durante o desenvolvimento do software é uma meta valiosa que precisa ser alcançada para reduzir o custo amplo do desenvolvimento.”

Levando em conta esse custo do desenvolvimento algumas empresas adotam o uso de framework híbrido como por exemplo o React Native, e segundo Sabino e Leão (2016) “aplicações híbridas são aplicações multiplataforma. Elas permitem, por exemplo, que você construa um produto iOS e Android simultaneamente, em vez de desenvolver duas vezes usando a linguagem nativa de cada plataforma”. O que pode ajudar a manter o orçamento dentro do estabelecido já que será necessário apenas uma ferramenta. Porém existe a possibilidade de fazer o desenvolvimento nativo usando Java/Kotlin para Android ou Objective-C/Swift para iOS.

Além do conflito financeiro mencionado, por se tratar de planejamento de software que envolve pessoal e trabalho em equipe intenso, Boehm (1984) afirma: “Atributos de pessoal e atividades de fornecimento de relações humanas são de longe a maior fonte de oportunidade para melhorar a produtividade do software”.

Portanto, como foi citado existe alguns fatores que devem ser levados em consideração no planejamento do desenvolvimento de software e analisando essa situação quais são as vantagens e desvantagens do framework React Native para a construção de software multiplataforma quando comparado a plataforma Nativa Android usando Java/Kotlin?

Objetivos

Objetivo Geral

Analisar as vantagens e desvantagens do framework React Native para desenvolvimento multiplataforma buscando auxiliar no planejamento dos projetos de software e discernir entre suas características os motivos para sua implementação em massa como opção forte e consistente de desenvolvimento.

Objetivos Específicos

- Analisar os custos para softwares multiplataforma;
- Identificar as limitações dos frameworks de desenvolvimento multiplataforma;
- Mensurar as ofertas de profissionais para as tecnologias híbridas;
- Investigar a manutenibilidade e suporte dos frameworks híbridos;
- Quantificar a produtividade do desenvolvimento híbrido;
- Identificar a oferta de API (*Application Programming Interface*) para as ferramentas de desenvolvimento híbrido;

Justificativa

A demanda no desenvolvimento de aplicativos vem crescendo consideravelmente e com a Covid-19 e isolamento social impulsionou criando um aumento significativamente no uso de dispositivos moveis.

As instalações de aplicativos pagos e orgânicos definitivamente aumentaram durante a pandemia; consumidores em todo o mundo instalaram cerca de 4 bilhões de aplicativos no primeiro semestre de 2020. Sem dúvida, eles estavam ocupando todo o tempo livre que ganhavam por não se deslocar para o trabalho ou escola ou participar de atividades pessoais. As instalações orgânicas e pagas aumentaram 21% e 15%, respectivamente, provando que quando as pessoas têm mais tempo livre, a melhor maneira de preenche-lo são os aplicativos em seus dispositivos (ABEINFO, 2021).

E um dos frameworks híbridos utilizado por grandes empresas é o React Native uma biblioteca desenvolvida com JavaScript criado em 2015 e usado atualmente no Instagram, Uber Eats, Discord, Tesla e etc. (Facebook, 2021). E como essa ferramenta é bastante difundida e com uma linguagem de programação entre as mais populares do mundo segundo Stack Overflow (2020) o JavaScript está em primeiro lugar em popularidade. E devido a esses fatores será usado como objeto de estudo o React Native.

O desenvolvimento Nativo será feito na plataforma Android um sistema operacional construído pelo Google em 2008 isso porque segundo a Statista (2021) os smartphones com o sistema operacional Android detêm 87% do mercado global em 2019 e espera-se que aumente nos próximos anos. O sistema operacional móvel desenvolvido pela Apple (iOS) tem uma participação de mercado de 13%. O que permite uma boa base de comparação com esse volume de usuários do Android e por isso será usado como objeto de estudo.

E com tantas opções de ferramentas e particularidades entre as plataformas como escolher a melhor para o seu projeto e contexto seja empresa ou profissional diante de tantas incógnitas este estudo visa colaborar evidenciando vantagens e desvantagens entre as ferramentas mais usadas no mercado para mostrar de maneira clara e objetiva dados relevantes para auxiliar na escolha e planejamento do desenvolvimento de software.

Fundamentação Teórica

Será feita a introdução dos itens que irão compor este trabalho no qual inclui plataformas híbridas e nativas, Java/Kotlin, React Native e os métodos utilizados para quantificar os esforços empregados em desenvolvimento de funcionalidades.

O desenvolvimento para plataformas híbridas consiste em fazer um código para funcionar nos principais sistemas operacionais. Já o nativo implica em construir aplicações apenas para um tipo de sistema.

Abordagens de desenvolvimento de plataforma híbrida oferecem aos desenvolvedores de aplicativos móveis a oportunidade de implementar aplicativo em várias plataformas móveis a partir de uma única base de código. Eles permitem escrever código uma vez e executá-lo em qualquer lugar. Sem essas abordagens, os desenvolvedores têm que desenvolver e manter bases de código separadas para cada plataforma que eles querem oferecer seus produtos. Essa abordagem é conhecida como desenvolvimento nativo. Atualmente, o mercado de telefonia móvel é dominado por duas plataformas Google Android e Apple iOS (DORFER; DEMETZ; HUBER, 2020).

O desenvolvimento nativo em Android usa Java e/ou Kotlin são duas linguagens de programação a Oracle sendo responsável pelo Java e a JetBrains pelo Kotlin e podem ser usadas simultaneamente no desenvolvimento de aplicações para Android usando a IDE Android Studio.

O Android Studio oferece compatibilidade total com o Kotlin. Assim, você pode adicionar arquivos Kotlin ao seu projeto e converter o código da linguagem Java em Kotlin. Em seguida, você poderá usar todas as ferramentas atuais do Android Studio com o código Kotlin, como preenchimento automático, verificador de lint, refatoração, depuração, entre outras (ANDROID, 2020).

O React Native é uma biblioteca JavaScript criada pelo Facebook em 2015. Tendo contribuições de indivíduos e empresas em todo mundo, incluindo Callstack, Expo, Infinite Red, Microsoft e Software Mansion (Facebook, 2021).

O método para quantificar esforço do desenvolvimento *Goal/Question/Metric* (Objetivo, Questão, Métrica) conhecido como GQM é utilizado para extrair dados relevantes durante o processo de criação de aplicações. O Dal'Osto (2003) afirma que existe três níveis

no GQM. Primeiro Nível Conceitual, uma meta é definida para um objeto de acordo com uma variedade de razões. Segundo Nível Operacional, são utilizadas questões para definir a melhoria e garantir a meta solucionada. Terceiro Nível Quantitativo, é associado um conjunto de dados às questões, afim de respondê-las de forma quantitativa.

Metodologia

O estudo em questão refere-se a uma pesquisa quantitativa e qualitativa com as informações mais relevantes possíveis para colaborar com as empresas e profissionais que vão escolher com qual tecnologia querem trabalhar. E para evidenciar os aspectos segue a descrição das etapas a serem desenvolvidas.

Para analisar a oferta de profissionais e os custos da construção do software seja multiplataforma ou nativa será analisado dados estatísticos sobre oferta de mão-de-obra e vagas de emprego e a média salarial dos profissionais de acordo com a senioridade para as tecnologias React Native e Java/Kotlin usando os dados disponíveis em algumas plataformas como Stack Overflow, Glassdoor, LinkedIn, Statista, repositórios do Github colocando de forma gráfica para fácil interpretação.

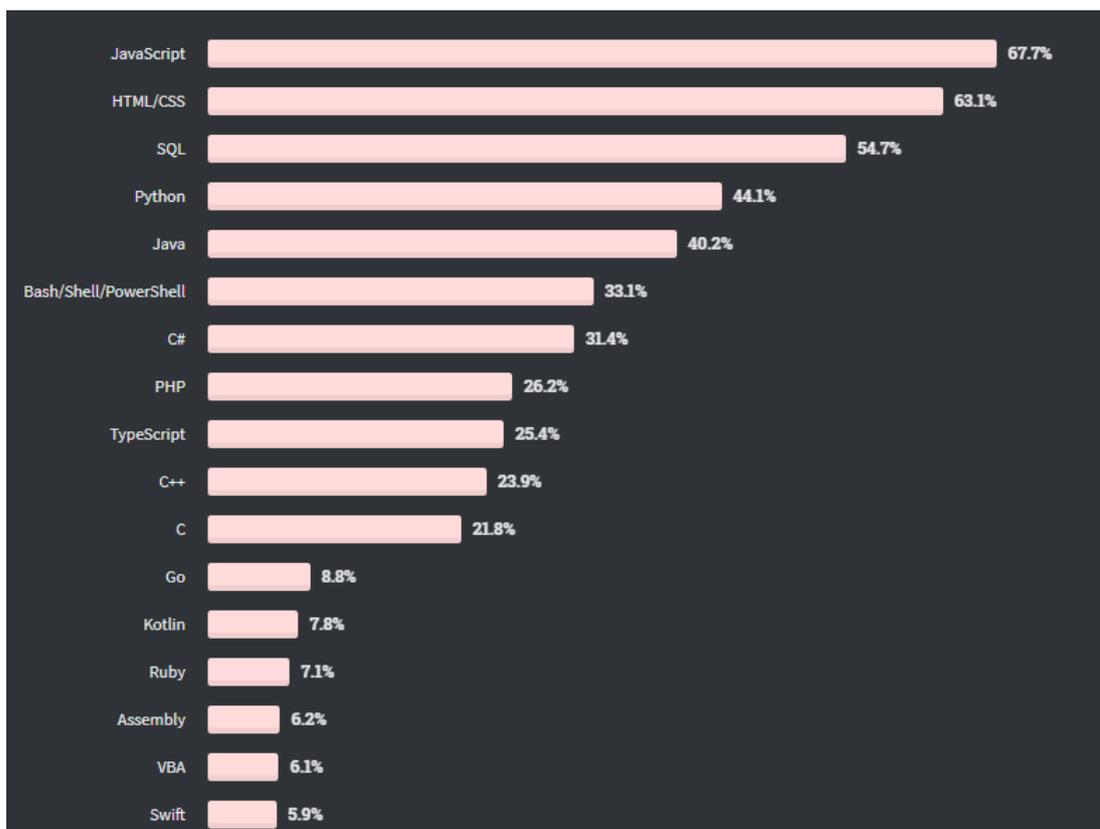
Já as limitações de cada ferramenta será feita uma análise descritiva sobre os recursos disponíveis para acessar as opções que hardware oferta para dispositivos mobile como câmera, sensores, leitor de digital entre outros verificando desempenho e características, usando como fonte de pesquisa as documentações oficiais dos frameworks e APIs disponíveis e a manutenibilidade dos mesmos para auxiliar nessas funcionalidades e os dados levantados serão estruturados em uma tabela seguindo com linhas listando os recursos dos dispositivos mobile e contendo as colunas: Desempenho/Características React Native e Desempenho/Características Java/Kotlin.

O desenvolvimento de funcionalidades para exemplificar pontos relevantes das ferramentas se faz necessário e para fazer a quantificação de esforço será usada o método GQM. E assim medir e evidenciar o esforço para o desenvolvimento da mesma função em diferentes plataformas. Nesse caso será desenvolvido um login com acesso a biometria com impressão digital assim podendo transmitir uma visão clara de acesso a um recurso que vem se tornando bastante popular. E assim extrair dados de tempo necessário para construção, documentação e suporte.

Resultados alcançados

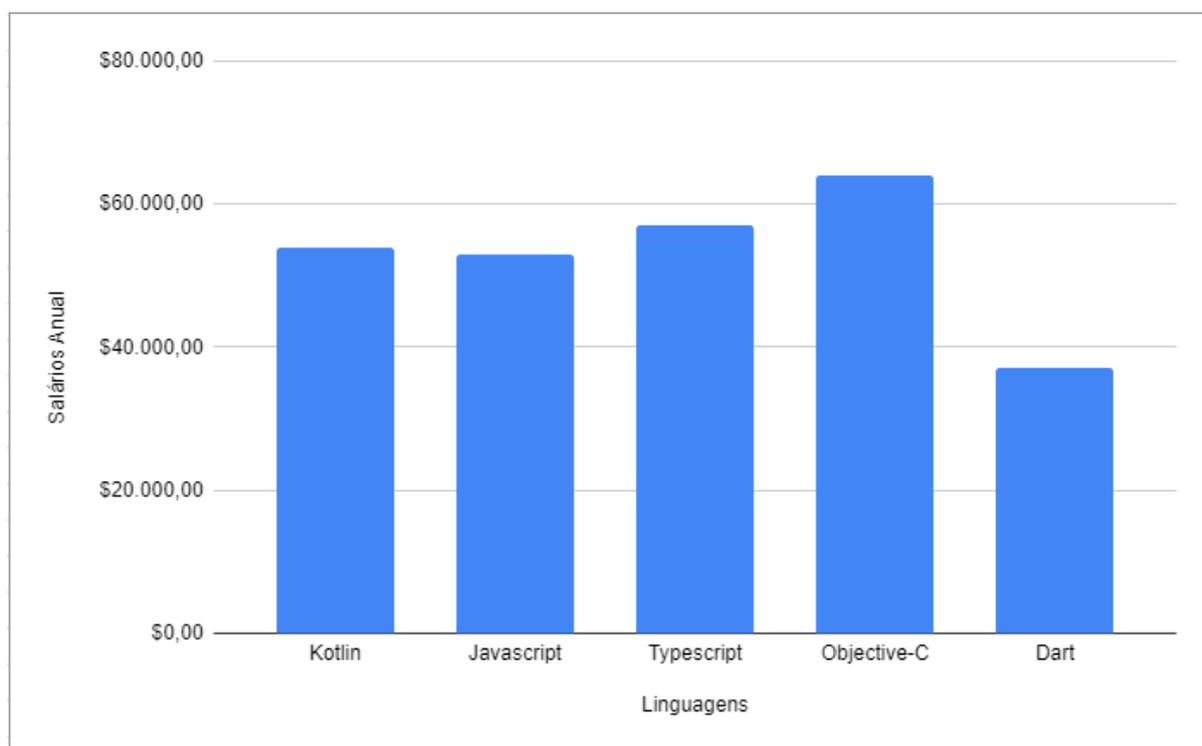
Um dos itens que é importante para analisar as vantagens e desvantagens de uma ferramenta é o custo do desenvolvimento e este valor é bastante variável e existe alguns fatores que influenciam diretamente como oferta de vagas, mão-de-obra, ferramentas e salários. Sobre oferta de vagas houve um aumento na demanda por profissionais na área de tecnologia da informação devido a Covid-19 o qual pode gerar um déficit de profissionais de até 260 mil vagas até 2024 segundo Sena e Granato (2021). E entre as habilidades mais desejadas pelas empresas estão Git, JavaScript e ReactJS (SENA, 2021). Segundo o Stack Overflow (2020) um dos maiores sites de perguntas e respostas sobre programação do mundo colocam o JavaScript como tecnologia mais popular em 2020 e também traz o salário como podemos ver no GRAF. 1 e GRAF. 2.

GRÁFICO 1 – Linguagens de programação, script e marcação



Fonte: Stack Overflow (2020)

GRÁFICO 2 – Linguagens e salários anuais

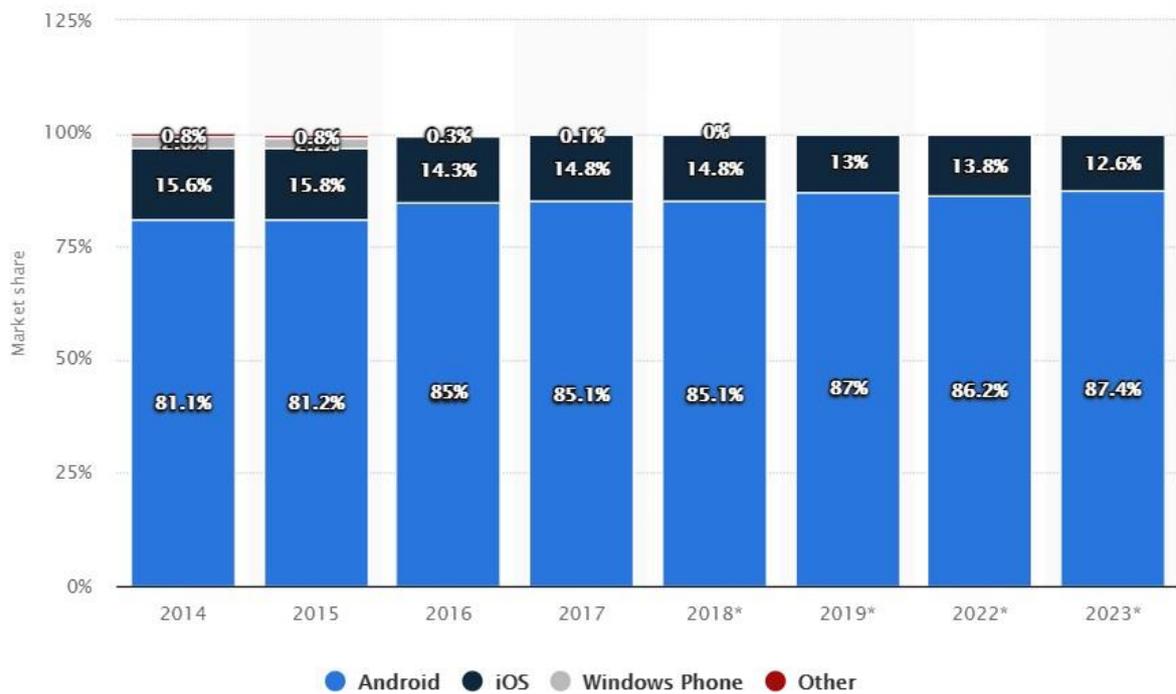


Fonte: Adaptado do Stack Overflow (2020)

E o número de desenvolvedores no mundo chega a 23,9 milhões e aqueles que atuam com JavaScript chega a ser mais da metade o que coloca o em destaque a facilidade para encontrar mão-de-obra para atuar com essa tecnologia. No terceiro trimestre de 2020, de acordo com a Developer Economics, o número de desenvolvedores de software usando JavaScript foi de 12,4 milhões. Isso significa que 53% de todos os desenvolvedores do mundo usaram JavaScript em algum momento (DAXX, 2020).

Outro fator relevante é número de usuários das principais plataformas mobile no mundo tendo hoje um cenário totalmente dominado em quantidade pelo Android que chega a ser 85% entre todos os dispositivos mobile de acordo com o GRAF. 3. O que favorece o uso de Java/Kotlin para atingir o maior número de usuários significativamente.

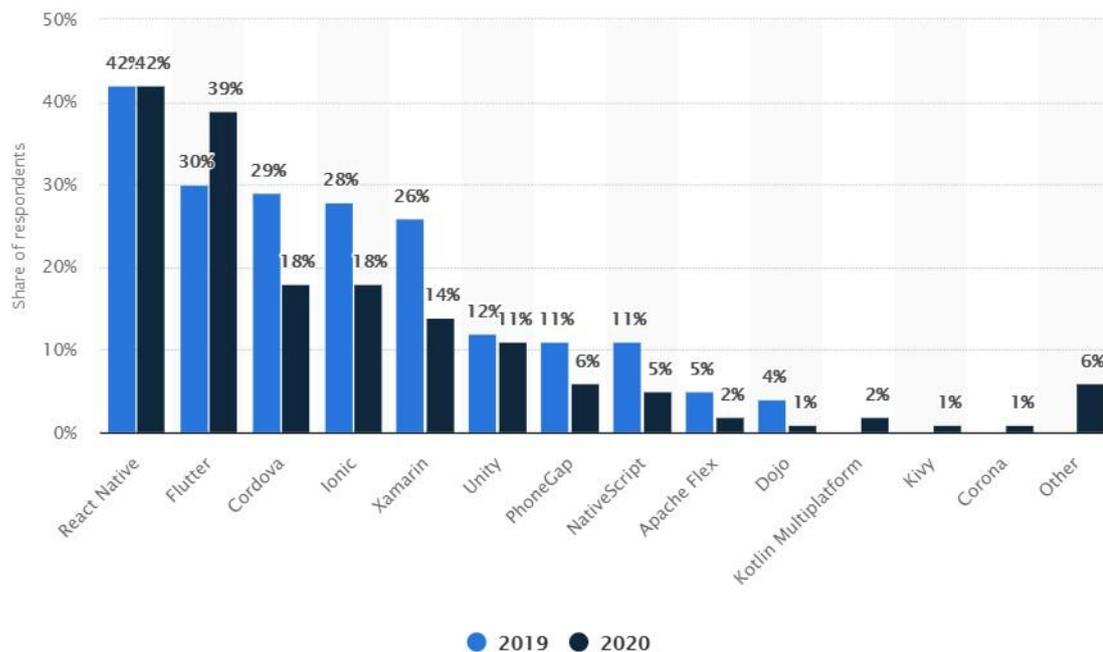
GRÁFICO 3 – Parcela das remessas globais de smartphones por sistema operacional de 2014 a 2023



Fonte: Statista (2021)

No cenário multiplataforma o React Native vem crescendo consideravelmente nos últimos anos e reforça o cenário das ferramentas que utilizam JavaScript para o desenvolvimento. Como podemos ver no GRAF. 4.

GRÁFICO 4 – Frameworks multiplataforma usado por desenvolvedores de software no mundo em 2019 e 2020



Fonte: Statista (2020)

Resultados esperados

O objetivo é quantificar os dados que são relevantes para demonstrar de forma eficiente as opções benéficas e os contrapontos de implementar o uso de framework híbrido em relação ao nativo, com foco no React Native e Java/Kotlin e mostrar as potencialidades de ambas e o crescimento de cada uma no mercado de tecnologia.

Impactar positivamente no âmbito do desenvolvimento de software colaborando com empresas e profissionais que podem usar esta pesquisa como uma fonte de referência para auxiliar a fazer a melhor escolha de tecnologia para desenvolver seus produtos. Colocando de forma clara as vantagens e desvantagens de cada escolha. E ajudar a diminuir o índice de fracassos das construções de softwares por incompatibilidade de problema de negócio e recursos ofertados pelas ferramentas.

Ampliar a gama de conhecimentos técnicos de desenvolvimento de software na atualidade relacionados aos métodos nativos e híbridos, com React Native e Java/Kotlin, buscando apresentar a idealização do desenvolvimento multiplataforma como uma forma de otimização de processos e métodos para uma empresa ou organização.

Demonstrar o crescimento da metodologia híbrida nos últimos anos com o objetivo de apresentar seu impacto positivo na atualidade. E analisar a sua projeção de crescimento a longo prazo usando os dados coletados. E prover uma visão crítica aos leitores deste estudo para que façam as devidas ressalvas quando estiverem em posições cruciais em relação ao planejamento do desenvolvimento de software.

Referências Bibliográficas

ABEINFO (São Paulo, Brasil). Associação de Empresas e Profissionais da Informação. *In: 2021: 5 tendências que vão ditar o sucesso dos apps no Brasil*. Barra Funda - SP, 28 jan. 2021. Disponível em: <https://abeinfo brasil.com.br/2021-5-tendencias-que-vao-ditar-o-sucesso-dos-apps-no-brasil/#:~:text=A%20descoberta%20de%20novos%20aplicativos,demanda%20reprimida%20por%20dispositivos%205G.&text=Nos%20primeiros%20tr% C3% AAs%20meses%20da,h% C3% A1%20um%20ano%20ou%20mais>. Acesso em: 3 maio 2021.

ANDROID. Adicionar o Kotlin a um app existente. *In: Adicionar o Kotlin a um app existente*. [S. l.], 28 dez. 2020. Disponível em: <https://developer.android.com/kotlin/add-kotlin#:~:text=Android%20Studio%20provides%20full%20support,refactoring%2C%20debugging%2C%20and%20more>. Acesso em: 4 maio 2021.

BOEHM, BARRY. SOFTWARE ENGINEERING ECONOMICS. *IEEE Transactions on Software Engineering*, [S. l.], p. 200-217, 1984. Disponível em: <http://bls.buu.ac.th/~s55103/00CourseResource/Boehm-SE-Economics.pdf>. Acesso em: 3 abr. 2021.

DAL ‘OSTO, Fábio. **Método para avaliação de ambientes de desenvolvimento de software combinando CMM e GQM**. Porto Alegre - RS, 2003. Disponível em: <https://lume.ufrgs.br/handle/10183/6818#:~:text=O%20modelo%20GQM%20descreve%20uma,e%20quest% C3% B5es%20que%20as%20satisfazem.&text=Esses%20grafos%20per mitem%20a%20visualiza% C3% A7% C3% A3o,suas%20metas%2C%20quest% C3% B5es%20e%20m% C3% A9tricas>. Acesso em: 10 maio 2021.

DAXX (Estados Unidos). **How Many Software Developers Are in the US and the World?** [S. l.], 9 fev. 2020. Disponível em: <https://www.daxx.com/blog/development-trends/number-software-developers-world>. Acesso em: 11 maio 2021.

DORFER, Thomas; DEMETZ, Lukas; HUBER, Stefan. Impact of mobile cross-platform development on CPU, memory and battery of mobile devices when using common mobile app features. *Procedia Computer Science*, [s. l.], v. 175, n. 2019, p. 189–196, 2020. Disponível em: <https://doi.org/10.1016/j.procs.2020.07.029>

FACEBOOK. React Native. *In: React Native*. [S. l.], 2021. Disponível em: <https://reactnative.dev>. Acesso em: 4 maio 2021.

SABINO, Felipe; LEÃO, Pedro. Híbrido vs Nativo. *In: Híbrido vs Nativo*. [S. l.], 16 out. 2016. Disponível em: <https://medium.com/taqtilebr/h% C3% ADbrido-vs-nativo-c8591df0dce6>. Acesso em: 3 maio 2021.

SENA, Victor; GRANATO, Luísa. 260.000 vagas sem dono: um raio-x das vagas mais quentes agora (e no futuro). *EXAME*, [S. l.], p. 1, 18 fev. 2021. Disponível em: <https://exame.com/carreira/260-000-vagas-de-trabalho-sem-dono-conheca-o-setor-que-ganhou-forca-com-a-pandemia/>. Acesso em: 10 maio 2021.

SENA, Victor. Veja 15 profissões em alta para 2021: saúde e tecnologia saem na frente. **Exame**, [S. l.], p. 1, 3 fev. 2021. Disponível em: <https://exame.com/carreira/veja-15-profissoes-em-alta-para-2021-saude-e-tecnologia-saem-na-frente/>. Acesso em: 11 maio 2021.

SINGHA , Tribhuvan; SINGHA, Ranvijay; MISHRAA, Krishn Kumar. Software Cost Estimation Using Environmental Adaptation Method. **Procedia Computer Science**, [S. l.], p. 1, 2018. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050918321008?via%3Dihub>. Acesso em: 13 abr. 2021.

STACK OVERFLOW. Programming, Scripting, and Markup Languages. In: Programming, Scripting, and Markup Languages. [S. l.], 2020. Disponível em: <https://insights.stackoverflow.com/survey/2020/#technology-programming-scripting-and-markup-languages>. Acesso em: 4 maio 2021.

STATISTA. Telecommunications. *In: Share of global smartphone shipments by operating system from 2014 to 2023*. [S. l.], 19 mar. 2021. Disponível em: <https://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems/#:~:text=Smartphones%20running%20the%20Android%20operating,percent%20share%20of%20the%20market>. Acesso em: 4 maio 2021.

STATISTA. Telecommunications. *In: Cross-platform mobile frameworks used by software developers worldwide in 2019 and 2020*. [S. l.], 2 jun. 2020. Disponível em: <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>. Acesso em: 4 maio 2021.