



Associação Propagadora Esdeva  
Centro de Ensino Superior de Juiz de Fora – CES/JF  
Curso de Engenharia de Software  
Projeto de Iniciação Científica

---

## Uma abordagem de aprendizado integrada e interdisciplinar através de um processo de desenvolvimento cross-platform

Júlia Condé Araújo<sup>1</sup>  
Marina Condé Araújo<sup>1</sup>  
Frâncila Weidt Neiva<sup>2</sup>

### RESUMO

O termo “cross-platforms” derivado do inglês significa “plataformas cruzadas”. Também chamados de frameworks híbridos, podem ser utilizados em mais de uma plataforma, ou seja, utilizando apenas um código o criador é capaz de portar em diferentes sistemas operacionais. O objetivo do trabalho foi avaliar, através de pesquisas, as semelhanças e as diferenças entre o desenvolvimento utilizando frameworks híbridos e nativos, além de comparar dentro dos “cross- platforms” quais são as vantagens e desvantagens dos principais frameworks híbridos.

**Palavras-chave:** framework multiplataforma, framework, híbrido, nativo

---

<sup>1</sup> Discente do Curso de Engenharia de Software do Centro de Ensino Superior de Juiz de Fora

<sup>2</sup> Docente do Curso de Engenharia de Software do Centro de Ensino Superior de Juiz de Fora

## 1 INTRODUÇÃO

Cross platform e Native frameworks são considerados importantes para aplicativos de desenvolvimento, embora sejam muito diferentes, ambos têm suas próprias vantagens e desvantagens.

Frameworks nativos são desenvolvidos usando uma linguagem / tecnologia para uma plataforma específica. Um framework nativo para Android não funcionará no IOS e vice-versa, por isso esse desenvolvimento é mais caro e demora mais para rodar, já que é preciso fazer mais de um código. Por outro lado, se o objetivo é trabalhar com ferramentas de dispositivo de baixo nível, como mensagens, acesso à câmera e até realizações de chamadas é recomendado usar os desenvolvimentos nativos.

Já os Cross platform ou frameworks híbridos podem ser utilizados em mais de uma plataforma, ou seja, utilizando apenas um código que o criador é capaz de portar em diferentes sistemas operacionais.

## 2 REFERENCIAL TEÓRICO

### 1- Flutter

O framework Flutter é o mais novo entre os quatro tratados e atualmente é considerado um dos principais para o desenvolvimento de aplicativos móveis. Criado pelo Google, através dele consegue desenvolver aplicativos nativos para Android e IOS a partir de uma única base de código. Apesar de ser um framework criado recentemente, várias empresas têm investido no Flutter para desenvolver suas soluções, assim, o framework tem tudo para se tornar uma excelente opção de estudo para desenvolvedores de software. Entre suas vantagens está o desenvolvimento de aplicativos móveis mais rápidos, por possuírem widgets próprios, os aplicativos possuem uma interface mais leve. Outro ponto é o Hot reload, que pode ser considerado um de seus melhores benefícios, pois é um recurso que permite ao desenvolvedor visualizar as alterações feitas no código imediatamente, economizando tempo, facilitando correções e evitando erros. Além disso, possui suporte oficial do Google, facilidade de configuração e menos código, possuem a mesma interface mesmo em aparelhos antigos. Entre suas desvantagens são apenas compatíveis com mobile, é um framework recente e possui poucas bibliotecas. Ele é usado em grandes empresas, bem como no Google, eBay, Square e BMW.

### 2- React – Native

A estrutura React-Native é uma das estruturas JavaScript mais populares da atualidade. É famoso por sua utilidade para a criação de aplicativos nativos nas plataformas Android e IOS. Esta estrutura encontra suas raízes no React, é uma biblioteca JavaScript que o Facebook criou para projetar a interface de aplicativos móveis. Entre suas vantagens está o desenvolvimento econômico, oferecendo uma forma mais barata de criar aplicativos multiplataforma com o React Native, já que pode implantar um único código para aplicativos IOS e Android. Além de ser uma forma mais rápida de realizar projetos, usando a linguagem de programação JavaScript, requer pessoal menor, também usa o sistema de código aberto e tem aparência de aplicativos nativos. Por outro lado, eles apresentam desafios de depuração e

compatibilidade, precisam de desenvolvedores nativos, dependem do Facebook, não têm uma memória excelente, têm problemas com licença e patente, adoção lenta de funções mais recentes, desafios de segurança do JavaScript. Entre seus usuários estão Facebook, Instagram, Tesla, Airbnb, Wix

### **3- Ionic**

O framework Ionic é uma plataforma que vem crescendo no mercado devido à sua alta produtividade e ao fácil manuseio de seus recursos já que muitos desenvolvedores aprendem desde cedo os conceitos utilizados pela plataforma. Por meio da combinação de JavaScript, HTML, CSS, Angular, TypeScript, JavaScript, Node e Cordova, o mesmo aplicativo pode ser disponibilizado nas plataformas Android, iOS e Windows Phone com comandos simples fornecidos pelo framework. Entre suas vantagens estão para empreendedores e donos de empresas, o desenvolvimento iônico pode ser benéfico se eles desejam desenvolver um aplicativo móvel em ambas as plataformas em um curto período de tempo em comparação com aplicativos nativos. Construir aplicativos nativos especificamente para cada plataforma pode ser demorado, o que pode implicar em um atraso no tempo de comercialização e o custo de desenvolvimento de aplicativos nativos é geralmente caro. Existem algumas desvantagens que vêm junto com o desenvolvimento do Ionic, como plug-ins nativos instáveis, depuração complicada, inadequada para desenvolvimento de jogos e segurança de aplicativos.

### **4- Xamarin**

O Xamarin é uma ferramenta utilizada para o desenvolvimento de aplicativos móveis multiplataforma que permitem aos engenheiros compartilhar cerca de 90% do código nas principais plataformas. Por ser uma ferramenta relativamente nova, é baseada na pilha de tecnologia da Microsoft e já possui uma comunidade de mais de 1,4 milhão de desenvolvedores. Entre as vantagens do framework estão uma pilha de tecnologia para codificar todas as plataformas, desempenho próximo ao nativo, experiências nativas do usuário, tecnologia de código aberto com forte suporte corporativo, suporte completo de hardware, manutenção simplificada, ecossistema de desenvolvimento completo, formulários Xamarin, que consiste de uma estrutura para aplicativos e protótipos simples e o reproduzidor ao vivo Xamarin, que simplifica

drasticamente todos os procedimentos de configuração e depuração. Suas desvantagens são o suporte ligeiramente atrasado para as atualizações de plataforma mais recentes, acesso limitado a bibliotecas de código aberto, alguns problemas com o ecossistema, conhecimento básico das línguas nativas necessárias, não adequado para aplicativos com gráficos pesados e problemas de compatibilidade de biblioteca. e ferramentas de terceiros.

<b>Critério</b>	<b>React Native</b>	<b>Ionic</b>	<b>Xamarin</b>	<b>Flutter</b>
Desenvolvedores	Facebook	Drifty	Microsoft	Google
Linguagem	JavaScript	Typescript	C#	Dart
Performance	Semelhante ao native	Moderado	Moderado	Amazing controlle
GUI – interface gráfica do usuário	Native UI	HTML, CSS	Native UI	Proprietary widgets
Testado	Dispositivos móveis	Qualquer navegador	Dispositivos móveis	Dispositivos móveis
Recarga	Sim	Não	Não	Sim

### 3 METODOLOGIA

No estudo de Kichtenham et al (2004) discute a possibilidade de adoção de pesquisas baseadas em evidências em engenharia de software, é frequentemente citada a revisão sistemática da literatura, uma espécie de estudo experimental que visa coletar evidências de estudos presentes na literatura para responder perguntas de pesquisas. Logo após essa discussão, Kitchenam e Charters (2004) publicaram um relatório técnico com instruções sobre como realizar uma revisão sistemática da literatura para a engenharia de software. Para encontrar métricas de complexidade estrutural dos programas, um planejamento sistemático da literatura foi planejado e executado. Seguindo as orientações de Kitchenam e Charters, o mapeamento permanece dividido em três etapas principais: (i) Planejamento do mapeamento, (ii) Realização do mapeamento e (iii) Resultados do mapeamento. Durante todas as etapas do mapeamento sistemático foi utilizada a ferramenta Parsifal, que permite que os pesquisadores em lugares geograficamente diferentes compartilhem o mesmo artigo, fornecendo uma plataforma que ajuda a gerenciar os estudos selecionados.

O planejamento do mapeamento para obter a revisão de literatura desejada foi dividido em quatro partes, sendo elas o PICOC, *String* de busca, critério de inclusão e exclusão e pergunta da pesquisa.

#### 1- PICOC

A busca de estudos durante o mapeamento ou revisão sistemática da literatura é regida por uma *String* de busca booleana, derivada de um conjunto de palavras – chave denominado PICOC, que visa delimitar o escopo da busca. Cada letra da sigla PICOC tem um significado na especialização da busca, o “P” vem de *population* delimita a população alvo da pesquisa, o “I” vem de *intervention* que determina quais serão as intervenções na população, o “C” é de *comparison*, quando deseja comparar métodos ou técnicas eles devem ser listados nessa categoria. O “O” de *outcome*, contém o que deseja analisar no final da pesquisa, resultados e por fim o “C” de *context*, responsável por limitar o contexto ao qual a pesquisa será limitada.

PICOC	KEYWORDS
Population	cross platform framework, cross-platform framework
Intervention	comparison, benchmark
Control	Not defined
Outcome	framework, pros, cons
Context	hybrid mobile software development

## 2- String de Busca

Juntando todos os termos do PICOC é possível criar a *string* de busca booleana, usada como base para derivar as *strings* específicas do mapeamento nas bases de busca digital.

```

("cross platform framework" OR "cross-platform framework") AND
("comparison" OR "benchmark") AND ("framework" OR ("cons" AND
"pros")) AND ("hybrid" AND "mobile" AND "software" AND
"development")

```

## 3- Critério de inclusão e exclusão

Após executar a string de pesquisa nas bases digitais de artigos, são apresentados todos os artigos que de acordo com a máquina de busca da base digital utilizada, respeitando as regras definidas na string de busca fornecida. Porém, o mecanismo de busca nem sempre retorna apenas artigos que são do interesse do pesquisador, ou seja, mesmo com a string de busca ajustada de acordo com as necessidades, são necessários critérios para definir quais estudos serão ou não aceitos para participar do mapeamento. Esses critérios são avaliados por meio da leitura do título e do resumo de todos os estudos retornados pelas bases de dados digitais de artigos.

	<b>Descrição</b>
Inclusão	Artigos contendo estudos experimentais, estudos de caso ou análises de qualquer cross platform framework
	Estudos contendo uma taxonomia ou levantamento de cross platform framework
Exclusão	Capítulos de livros, chamadas a congressos e materiais de ensino médio
	Estudos que não podem ser acessados por inteiro
	Estudos que não abordem o tema na área de Engenharia de Software

#### 4- Pergunta da Pesquisa

As questões de pesquisa são cruciais no protocolo de mapeamento sistemático, pois é o fator que orienta o escopo da pesquisa.

Quais são os pros e contras entre as cross platform framework ?
---

#### 4 RESULTADOS E DISCUSSÃO

Após definir todos os detalhes do protocolo, é possível realizar o mapeamento sistemático. No contexto desta pesquisa em particular, são realizadas as seguintes atividades: (i) importação de estudos encontrados em bases de dados digitais, (ii) artigos duplicados e (iii) critérios de inclusão / exclusão.

A amostra foi composta por 61 artigos, sendo 1 duplicado, 20 removidos pelo critério de inclusão e exclusão e 40 artigos foram utilizados na pesquisa.

Importação de estudos encontrados em bases de dados digitais
61 artigos

Artigos duplicados
1 artigo

Critério de Inclusão/Exclusão
20 artigos

Artigos utilizados na pesquisa
40 artigos

## **5 CONSIDERAÇÕES FINAIS**

Pudemos perceber e compreender as diferenças e semelhanças entre os principais cross- platform frameworks e sua comparação com o native framework, tendo em vistas suas características é evidente que os desenvolvedores de software tem que estar cientes sobre as aplicações híbridas e nativas, para melhorarem a funcionalidade de seus projetos, visando seu público e o ambiente de funcionamento.

**ABSTRACT**

The term "cross-platform" derived from English means "cross-platforms". Also called hybrid frameworks, they can be used on more than one platform, that is, using just one code the creator is able to port on different operating systems. The objective of the work was to evaluate, through research, the similarities and differences between the development using hybrid and native frameworks, as well as to compare within "cross platforms" what are the advantages and disadvantages of the main hybrid frameworks.

**Keywords:** cross platform framework, framework, hybrid, native

## REFERÊNCIAS

1. Heitkötter, H., Kuchen, H., Majchrzak, T.A.: Extending a model-driven cross-platform development approach for business apps. *Sci. Comput. Program. (SCP)* 97(Part 1), 31–36 (2015)
2. Ohrt, J., Turau, V.: Cross-platform development tools for smartphone applications. *IEEE Comput.* 45(9), 72–79 (2012)
3. Smutny, P.: Mobile development tools and cross-platform solutions. In: *Proceedings of 13th ICC*, pp. 653–656 (2012)