

---

**COMPARAÇÃO ENTRE O DESEMPENHO DE APLICAÇÕES PARA  
SMARTPHONES DESENVOLVIDAS EM FLUTTER E REACT NATIVE: UMA  
ANÁLISE UTILIZANDO ALGORITMOS DE ORDENAÇÃO**

**COMPARISON BETWEEN THE PERFORMANCE OF SMARTPHONE  
APPLICATIONS DEVELOPED IN FLUTTER AND REACT NATIVE: AN ANALYSIS  
USING SORTING ALGORITHMS**

Carine Casa Grande <sup>1</sup>  
Simone Sawasaki Tanaka <sup>2</sup>

**RESUMO**

O uso de *smartphones* tem crescido consideravelmente nos últimos anos, tornando o mercado de desenvolvimento de aplicativos cada vez mais forte. Atualmente, existem dois principais sistemas operacionais para esses dispositivos, o iOS e o Android. Entretanto, como os dois possuem linguagens nativas diferentes, sendo o Swift e o Objective-C para o primeiro e Java e Kotlin para o segundo, o desenvolvimento nativo se mostra de alto custo. Alguns *frameworks* podem ser utilizados para desenvolver de forma híbrida um aplicativo, isto é, um único código que gera aplicativos para mais de um sistema operacional, diminuindo este custo e tornando o desenvolvimento mais prático. Este trabalho analisa e compara o desempenho dos aplicativos híbridos criados em Flutter e em React Native, dois dos principais *frameworks* que possuem tal objetivo. Para isso, foi criado um aplicativo para executar algoritmos de ordenação, onde o tempo de execução foi medido e os resultados comparados para cada sistema operacional. Embora trabalhos correlatos tenham determinado o Flutter como o *framework* que entrega o aplicativo com maior desempenho, os resultados mostram que o React Native se mostrou mais ágil nos testes realizados com os algoritmos de ordenação.

**Palavras-chave:** flutter; react native; desenvolvimento híbrido; *framework*.

**ABSTRACT**

The use of smartphones has grown considerably in recent years, making the application development market stronger and stronger. There are currently two main operating systems for these devices, iOS and Android. However, as the two have different native languages, being Swift and Objective-C for the first and Java and Kotlin for the second, native development proves to be costly. Some frameworks can be used

---

<sup>1</sup> Graduanda do Curso de Ciência da Computação do Centro Universitário Filadélfia - UniFil (carinecasagrandee@edu.unifil.br).

<sup>2</sup> Orientador: Professora do curso de Ciência da Computação do Centro Universitário Filadélfia - UniFil (simone.tanaka@unifil.br).

to hybrid develop an application, that is, a single code that generates applications for more than one operating system, reducing this cost and making development more practical. This work analyzes and compares the performance of hybrid applications created in Flutter and React Native, two of the main frameworks that have this objective. For this, an application was created to run sorting algorithms, where the execution time was measured, and the results compared for each operating system. Although related works have determined Flutter as the framework that delivers the application with the highest performance, the results show that React Native proved to be more agile in tests performed with the sorting algorithms.

**Keywords:** flutter; react native; hybrid development; framework.

## 1 INTRODUÇÃO

O desenvolvimento de aplicações móveis é uma área necessária no mercado de programação por conta da alta demanda que possui e, para atender a maior parte deste mercado, é preciso considerar os principais sistemas operacionais (SO) disponíveis para este tipo de dispositivo: o iOS e o Android (STATCOUNTER, 2021).

Entretanto, esses SO diferem entre si, o que torna o desenvolvimento nativo caro e mais demorado. De modo a agilizar o desenvolvimento e facilitar a manutenção dos aplicativos (*app*) foram criadas tecnologias que permitem o desenvolvimento híbrido, isto é, um único desenvolvimento que atenda a diversos sistemas.

Dentre essas tecnologias, podemos citar o Ionic, o Xamarin, o React Native e o Flutter como exemplo, sendo que as duas últimas serão abordadas no trabalho por serem as mais discutidas nos últimos tempos.

Desta forma, este trabalho visa comparar o Flutter e o React Native com base no desempenho dos mesmos durante a execução de algoritmos de ordenação de modo a determinar a melhor alternativa de *framework* para ser utilizada no desenvolvimento híbrido.

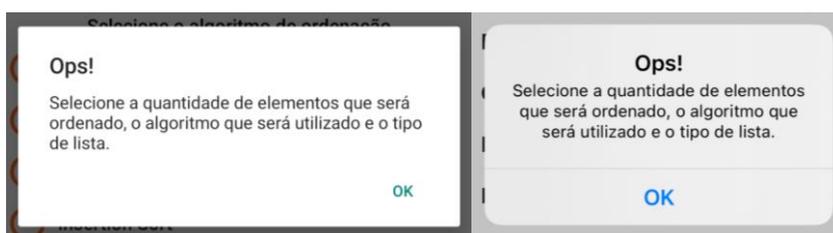
## 2 REACT NATIVE

De acordo com Leitão (2019), o React Native é um *framework* de código aberto utilizado na criação de aplicativos que tem como base o JavaScript (JS). Seu método de desenvolvimento é o JavaScript *Runtime*, ou seja, seu código é transformado em

nativo em tempo de execução e tem acesso aos recursos do dispositivo através de interfaces de programação de aplicações (API) nativas do sistema (GUEDES, 2020).

Um exemplo prático desta característica é a diferença entre a aparência de um componente em cada SO. Na Figura 1 é possível verificarmos como os estilos de borda, fonte e centralização do texto mudam em cada plataforma.

**Figura 1** — Elemento em React Native no Android (direita) e no iOS (esquerda).



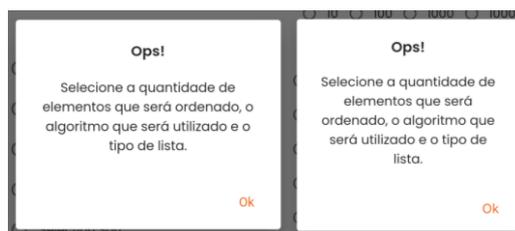
Fonte: Autor (2021)

### 3 FLUTTER

O Flutter é um *framework* de desenvolvimento híbrido baseado em Dart anunciado pela Google em 2015 e teve a primeira versão estável lançada em 2018 (BUENO, 2021). Ao criar um aplicativo utilizando o Flutter o mesmo é compilado para a linguagem nativa do sistema operacional, fazendo com que o mesmo possua acesso direto aos recursos do dispositivo (ANDRADE, 2020).

Uma característica do Flutter é que ele conta com *widgets* próprios, o que permite obter um *design de interface* personalizado (CANGUÇU, 2019), sendo uma das principais vantagens em comparação ao React Native. Isto pode ser visto na Figura 2, onde a aparência de um componente é a mesma tanto no Android (posicionado à direita) quanto no iOS (posicionado à esquerda): mesmo estilo de borda, fonte, etc.

**Figura 2** — Elemento em Flutter no Android (direita) e no iOS (esquerda).



Fonte: Autor (2021)

#### **4 REACT NATIVE VS. FLUTTER**

Por conta dos aplicativos desenvolvidos em Flutter serem compilados para a linguagem nativa do sistema em tempo de compilação, ele tende a obter os melhores resultados quando comparado com os aplicativos desenvolvidos em React Native.

Os testes de Demedyuk e Tsybulskyi (2020) apontam que o Flutter obteve um desempenho significativamente superior ao do React Native ao executar algoritmos que exigem memória e também em algoritmos que utilizam a unidade central de processamento (CPU).

O Flutter utilizou 2% a menos da CPU do que o React Native nas pesquisas de Stender e Åkesson (2020), entretanto, apesar da vantagem no consumo de CPU, o Flutter demorou mais para executar os códigos.

Quando comparado os dois *frameworks* em diversos critérios, como desempenho, plataformas, distribuição, viabilidade de longo prazo e outros, o React Native pode ser superior ao Flutter, mas por pouca diferença entre eles (HJORT, 2020).

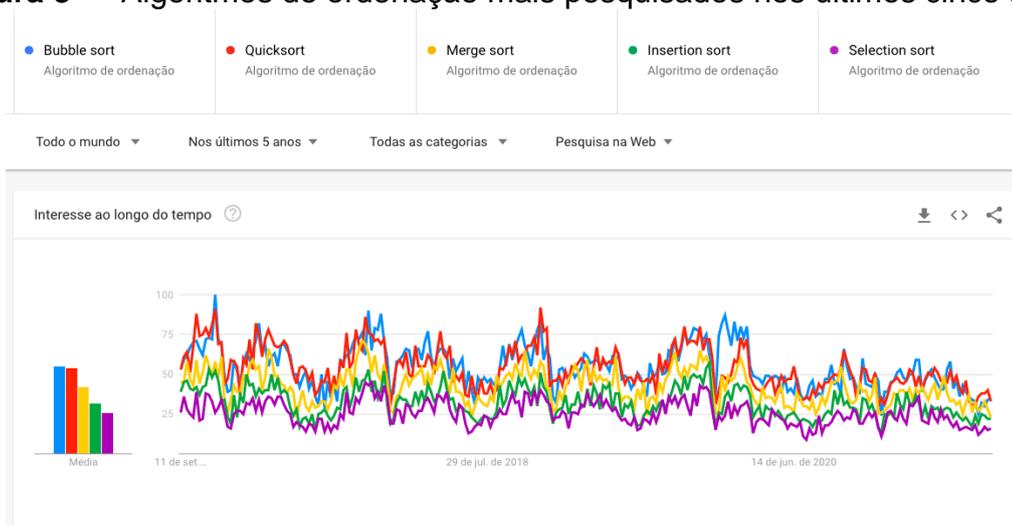
10

#### **5 ALGORITMOS DE ORDENAÇÃO**

Os algoritmos de ordenação são aqueles que visam colocar os elementos de uma dada sequência em certa ordem (HONORATO, 2013). Borin (2020) afirma que a busca pelo método de ordenação mais eficiente é um dos problemas principais no desenvolvimento de algoritmos.

Neste trabalho, foram considerados os algoritmos de ordenação mais pesquisados na ferramenta Google nos últimos cinco anos conforme o Google Trends (2021), como podemos observar na Figura 3.

**Figura 3** — Algoritmos de ordenação mais pesquisados nos últimos cinco anos.



Fonte: Google Trends (2021)

## 5 METODOLOGIA

De modo a atingir os objetivos do trabalho, foram utilizadas duas aplicações, uma desenvolvida em Flutter e uma em React Native, onde o usuário pode escolher o tamanho da lista que seria utilizada, o algoritmo de ordenação que seria executado e o tipo de lista. As listas podiam ser do tipo aleatória ou pré determinada, onde a primeira era gerada de forma aleatória sem repetir elementos e a segunda teve seus elementos previamente determinados.

11

No desenvolvimento dos aplicativos só foram utilizados códigos dos *frameworks*, sem que alterações nos códigos nativos fossem feitas, para que a comparação levasse em conta também o que cada framework entrega. Foi contabilizado o tempo de execução e os dados obtidos foram analisados e comparados entre os aplicativos nos sistemas operacionais para smartphones iOS e Android. As especificações dos dispositivos utilizados para fazer os testes podem ser vistas na Tabela 1.

**Tabela 1** – Especificação dos dispositivos utilizados para realizar os testes.

Modelo	Marca	Processador	Memória RAM	SO
Poco X3 Pro	Xiaomi	Snapdragon 860 Qualcomm	6GB	Android 11
iPhone XR	Apple	Apple A12 Bionic	3GB	iOS 15

Fonte: Tudo Celular (2021) (Adaptado pelo Autor, 2021)

## 7 DESENVOLVIMENTO

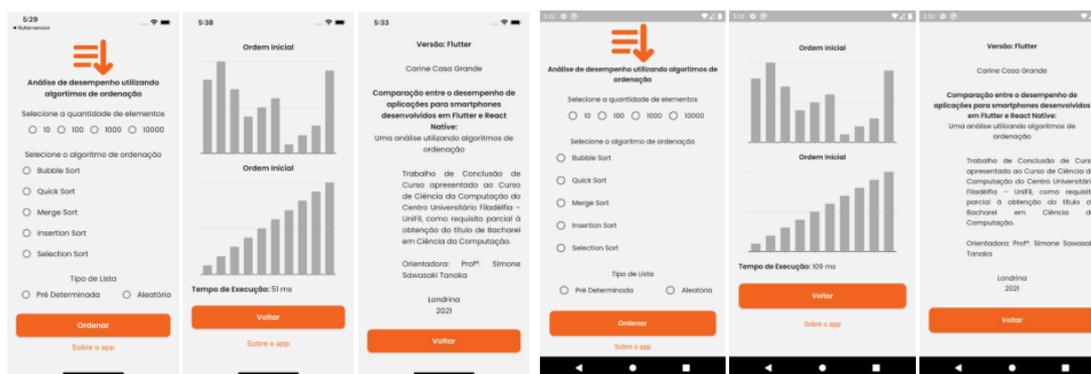
O desenvolvimento do aplicativo foi feito utilizando um MacBook Air 2017 da Apple com SO macOS Big Sur na versão 11.6. Além disso, o Visual Studio Code foi utilizado como editor de código-fonte e os aplicativos Android Studio e Xcode foram necessários para executar os simuladores durante o desenvolvimento. A configuração dos *frameworks* foi feita seguindo as suas respectivas documentações.

### 7.1 Desenvolvimento em Flutter

O desenvolvimento do aplicativo em Flutter demorou cerca de vinte horas, sendo que o desenvolvedor já possuía um conhecimento básico sobre o framework. Todas as dúvidas que surgiram ao decorrer do desenvolvimento foram facilmente sanadas com as respostas encontradas em fóruns da comunidade. Na Figura 4 é possível ver o resultado obtido.

12

**Figura 4** — Resultado do *app* construído em Flutter no iOS (esquerda) e no Android (direita).



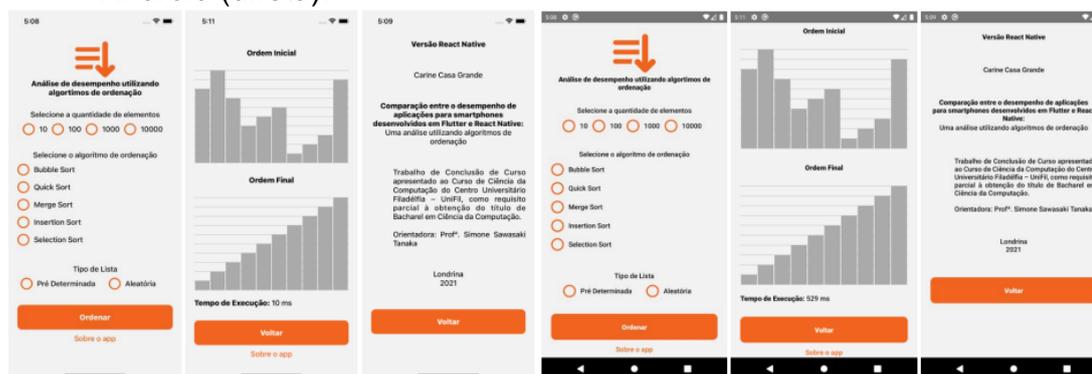
Fonte: Autor (2021)

### 7.2 Desenvolvimento em React Native

Para chegar no resultado, o desenvolvimento do aplicativo em React Native também demorou cerca de vinte horas, mesmo que o desenvolvedor nunca tivesse usado o React ou o React Native. Além disso, assim como no desenvolvimento do

app em Flutter, todas as dúvidas foram resolvidas com as respostas encontradas em fóruns da comunidade.

**Figura 5** - Resultado do app construído em React Native no iOS (esquerda) e no Android (direita).



Fonte: Autor (2021)

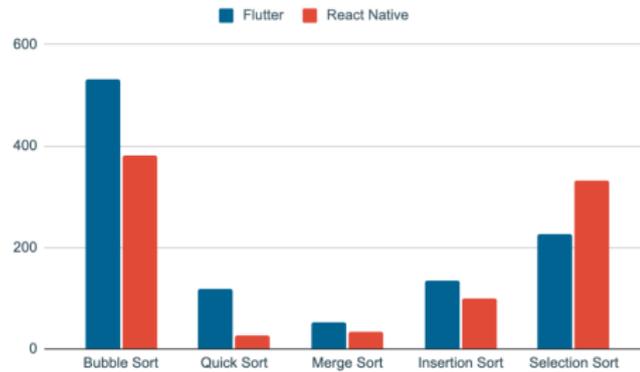
## 8 RESULTADOS E DISCUSSÃO

Nesta seção será abordado os resultados obtidos em cada sistema operacional, determinando o *framework* que obteve os melhores resultados em cada teste executado.

### 8.1 Android

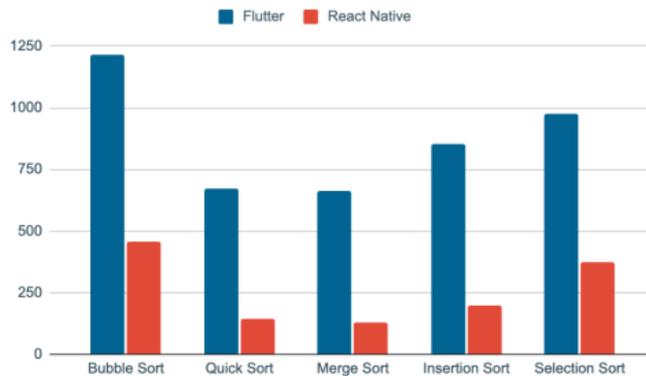
Na Figura 6, vemos que a média de tempo que o Flutter leva para executar os algoritmos na lista pré-determinada é inferior ao React Native em apenas um algoritmo além de que na lista aleatória, a diferença de tempo é discrepante, sendo o React Native mais vantajoso, como pode ser observado na Figura 7.

**Figura 6** – Média de tempo utilizando a lista pré-determinada por algoritmo de ordenação.



Fonte: Autor (2021)

**Figura 7** – Média de tempo utilizando a lista aleatória por algoritmo de ordenação.

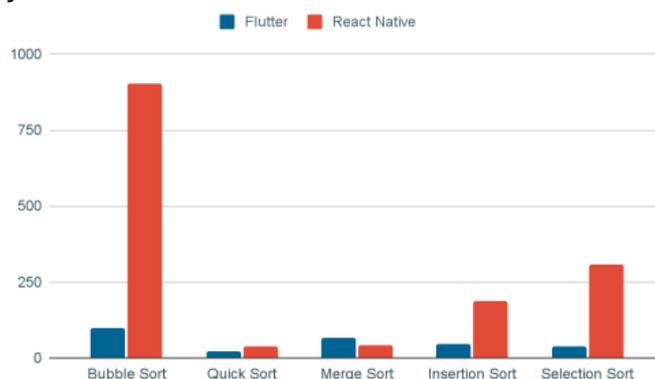


Fonte: Autor (2021)

## 8.2 iOS

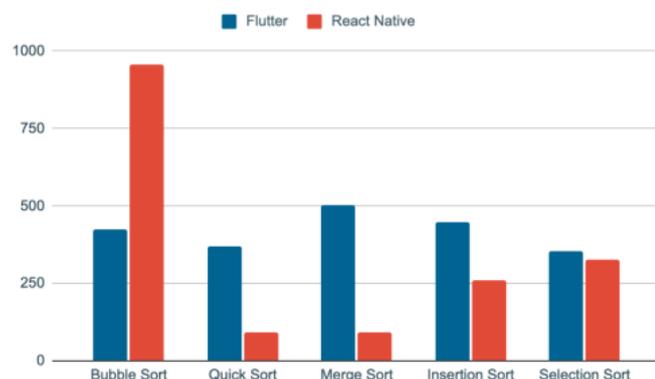
Ao comparar os resultados da lista pré-determinada no iOS, vemos que o resultado do Flutter foi melhor do que o resultado do React Native na maior parte dos casos, como podemos ver na Figura 8. Já ao analisarmos o resultado da lista aleatória na Figura 9, o Flutter obteve uma melhor média apenas na execução do Bubble Sort.

**Figura 8** – Média de tempo utilizando a lista pré-determinada por algoritmo de ordenação.



Fonte: Autor (2021)

**Figura 7** – Média de tempo utilizando a lista aleatória por algoritmo de ordenação.



Fonte: Autor (2021)

## 9 CONCLUSÃO

Este trabalho comparou o desempenho do Flutter e do React Native ao executar alguns algoritmos de ordenação, tendo como hipótese a ideia de que o Flutter teria os melhores resultados por conta dos trabalhos correlatos que existem na área. Entretanto, os testes apontaram que o React Native obteve os melhores resultados em 75% dos cenários testados, tendo executado os algoritmos em um tempo menor do que o Flutter. Além disso, o tempo de carregamento da página que o React Native forneceu foi inferior ao do Flutter, tornando o aplicativo mais fluido.

O fato de o Flutter ter obtido um resultado inferior não o torna uma tecnologia ruim, sendo que tudo depende da necessidade do aplicativo que será desenvolvido. O React Native se mostra uma tecnologia muito forte, tanto pelo seu desempenho

quanto pela facilidade que é aprender a tecnologia, entretanto muitos elementos devem ser importados, o que pode dificultar o desenvolvimento.

## REFERÊNCIAS

ANDRADE, Ana Paula de. **O que é Flutter?** Disponível em:

<https://www.treinaweb.com.br/blog/o-que-e-flutter>. Acesso em: 31 ago. 2021.

BUENO, Carlos Eduardo de Oliveira. **Desenvolvimento de um aplicativo utilizando o framework flutter e arquitetura limpa**. 2021. 47 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Pontifícia Universidade Católica de Góias, Goiânia, 2021.

CANGUÇU, Raphael. **React Native vs. Flutter**. Disponível em

<https://codificar.com.br/react-native-vs-flutter/>. Acesso em: 01 set. 2021.

DEMEDIYUK, Ihor; TSYBULSKYI, Nazar. **Flutter vs Native vs React-Native: Examining Performance**. Disponível em: <https://inveritasoft.com/blog/flutter-vs-native-vs-react-native-examining-performance>. Acesso em: 02 set. 2021.

FLUTTER. **Apps take flight with Flutter**. Disponível em <https://flutter.dev/showcase>. Acesso em: 01 set. 2021.

GUEDES, Marylene. **React Native ou Flutter: por qual começar?** Disponível em: <https://www.treinaweb.com.br/blog/react-native-ou-flutter-por-qual-comecar/>. Acesso em: 01 set. 2021.

HJORT, Elin. **Evaluation of React Native and Flutter for cross-platform mobile application development**. 2020. 67 f. Disponível em: <https://www.doria.fi/handle/10024/180002>. Acesso em: 02 set. 2021.

IBGE EDUCA. **Uso de internet, televisão e celular no Brasil**. Disponível em: <https://educa.ibge.gov.br/jovens/materias-especiais/20787-uso-de-internet-televisao-e-celular-no-brasil.html>. Acesso em: 31 ago. 2021.

LEITÃO, Paulo Rogério Matheus Bonfim. **Desenvolvimento nativo vs React Native: uma análise comparativa na codificação de uma aplicação para fitness**. 2019. 54 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas e Mídias Digitais) - Universidade Federal do Ceará, Fortaleza, 2019.

STATCOUNTER. **Mobile Operating System Market Share Worldwide: Jan - Dec 2020**. Disponível em: <https://gs.statcounter.com/os-market-share/mobile/worldwide/2020>. Acesso em: 31 ago. 2021.

STENDER, Simon; ÅKESSON, Hampus. **Cross-platform Framework Comparison: Flutter & React Native**. 2020. 59 f. Disponível em: <https://www.diva-portal.org/smash/record.jsf>. Acesso em: 02 set. 2021.

WU, Wenhao. **React Native vs Flutter, cross-platform mobile application frameworks**. 2018. 34 f. Disponível em: <https://www.theseus.fi/bitstream/handle/10024/146232/thesis.pdf>. Acesso em: 05 set. 2021.