

ENTRAR

MATRICULE-SE

TODOS OS  
CURSOSNOSSAS  
FORMAÇÕESPARA  
EMPRESASDEV  
EM <T>Artigos > **Mobile**

# Como fazer a migração do Expo para React Native CLI



**andre-cunha5**  
15/09/2021

COMPARTILHE

Tá pensando em começar a [programar em React Native](#) mas não sabe se começa usando **Expo CLI** ou **React Native CLI**? Ou já começou um projeto Expo e quer migrar ele para ter acesso a mais recursos e APIs nativas?

Então, você está no lugar certo! Nesse artigo abordaremos as principais diferenças, vantagens e desvantagens do Expo CLI e React Native CLI, além de mostrar como fazer a migração deles de uma maneira bem tranquila.

Vamos lá?

# Tópicos

Neste artigo, vamos ver:

- O que é CLI?
- O que é o Expo?
- Diferenças entre o Expo e o React Native CLI
- Migração do projeto Expo para o React Native CLI
- Conclusão

## O que é CLI?

Antes de falar do Expo CLI e o React Native CLI, você deve se perguntar: mas o que é CLI?

**CLI** é uma sigla em inglês que significa ***command-line interface*** ou **interface de linha de comando**. Basicamente, é uma forma de processar comandos que serão enviados para um programa de computador na forma de linhas de texto.

Atualmente, com interfaces gráficas de usuário (GUI), a maioria dos usuários não usa mais interfaces de linha de comando (CLI). Mas, ainda usamos CLIs para configurar computadores, instalar softwares e acessar recursos que não estão disponíveis na interface gráfica, como veremos em algumas situações com o Expo e o React Native puro.

Agora que aprendemos o que é CLI, podemos falar sobre o Expo CLI e React Native CLI. Primeiramente, vamos falar do Expo.

## O que é o Expo?

O **Expo** é uma ferramenta que facilita no desenvolvimento de aplicativos mobile com **React Native**, já que ele abstrai todas as partes complexas de configuração do ambiente e te permite acesso rápido e fácil a várias APIs nativas.

Com o Expo, é possível usar a API de câmera e notificações, por exemplo, sem muita dificuldade, já que não é necessário fazer nenhuma configuração de API.

Além disso, é possível programar para iOS sem precisar de um macOS, apenas um dispositivo físico com iOS. Isso tudo é possível porque a Expo disponibiliza um aplicativo móvel, que pode ser baixado pelas lojas da Play Store ou Apple Store, que já contém todo o código nativo para iniciar a aplicação React Native.

Com essas facilidades, o programador só precisa basicamente se preocupar com o código Javascript, sem ter que instalar SDK do Android ou Xcode para iOS. Dessa forma, em poucos minutos já é possível ver uma aplicação rodando diretamente no celular ou em um emulador ou simulador.

## Diferenças entre o Expo CLI e o React Native CLI

Bom, já vimos do que se trata o CLI e o Expo. Sabemos, também, que existe o React Native CLI.

Mas, afinal, qual a diferença do Expo para a interface de linha de comando do React Native?

A grande diferença entre usar o Expo CLI em relação ao React Native CLI é que, se você deseja integrar uma biblioteca nativa de terceiros ou estender a funcionalidade do aplicativo com código nativo, **você simplesmente não consegue fazer isso no Expo.**

Isso porque os projetos da Expo não revelam os códigos nativos iOS e Android. Ou seja, podemos dizer que o Expo é mais fácil de utilizar, porém tem algumas limitações: não podemos integrar uma biblioteca nativa no Expo, por exemplo.

Ainda assim, a maioria dos casos de uso para aplicativos pode ser coberta pela Expo CLI e os desenvolvedores sempre estão atualizando as funcionalidades nativas para que elas possam ser usadas de uma forma bem mais simples do que se você fosse configurar manualmente em aplicação React Native CLI. Mas, um exemplo de funcionalidade que ainda não está disponível, e os desenvolvedores da Expo estão trabalhando nisso, é usar o bluetooth. **Se o seu App precisa usar o bluetooth, será necessário fazer um projeto sem o Expo.**

Então, isso basicamente significa que você está limitado ao nível de funcionalidade nativa fornecida pelo React Native e Expo SDK. Isso pode ser um pouco ruim, porque novas funcionalidades lançadas pelo React Native demoram até serem implementadas no Expo.

Vejamos algumas vantagens e desvantagens de cada um:

# Expo CLI:

## Prós:

- facilidade em criar, iniciar e testar a aplicação em um aparelho físico ou virtual;
- configuração de ambiente tranquila;
- acesso simples à features como câmera, mapas, notificações e entre outras.

## Contras:

- não estão disponíveis todas APIs do Android e iOS;
- versões mais recentes do React Native e novas funcionalidades demoram um pouco mais a chegar para o Expo;
- alguns tipos de execução em segundo plano ainda não estão disponíveis;
- códigos nativos para iOS e Android não funcionam no Expo.

# React Native CLI:

## Prós:

- são suportados códigos nativos para Android e iOS;
- altamente customizável;
- facilidade para atualizar a versão mais recente do React Native, para aproveitar todos os recursos novos disponíveis.

## Contras:

- complicada configuração inicial do ambiente;
- é necessário lidar com todas as configurações, assinaturas e certificados para enviar o aplicativo para produção.

Agora que você já sabe sobre as vantagens e desvantagens do Expo e do React Native CLI, é hora de decidir qual vale mais a pena para sua aplicação. Se quiser, compartilhe suas ideias com a gente no fórum!

# Migração do projeto Expo para o React Native puro

Beleza, mas já comecei meu projeto com Expo e percebi que vou precisar de um recurso que ainda não está disponível nele. Vou precisar começar do zero no React Native CLI?

Por sorte, não! O Expo possui um recurso que nos ajuda nisso. Chamamos isso de **ejeção** e é muito simples fazer isso. Veja o passo a passo abaixo:

1. Primeiro, navegue até a raiz do seu projeto Expo;
2. Abra o terminal na pasta do projeto e digite: `expo eject`;
3. Ele vai criar duas pastas: uma pasta android e outra ios e alguns outros arquivos;
4. Quando todo o processo terminar, você já terá sua aplicação migrada para o React Native CLI.

Tá, mas como eu executo ela? É só digitar “expo start” ou “npm start”? Infelizmente, não é tão simples. Alguns passos a mais são necessários para rodar a aplicação.

É importante falar que, para executar os passos abaixo, é preciso ter o ambiente de desenvolvimento configurado. É possível conferir isso nesse [artigo](#).

Se você já tem o ambiente de desenvolvimento configurado, vamos lá, então!

Abra o seu emulador Android ou simulador iOS e já o deixe ligado e rodando. Para executar a aplicação, será necessário usar 2 terminais. Nos dois terminais, navegue até dentro da pasta do projeto.

No terminal 1, digite:

```
npm start
```

No terminal 2, digite:

Se você estiver usando um emulador Android:

```
npx react-native run-android
```

Ou caso seja um simulador iOS:

```
npx react-native run-ios
```

Pronto, sua aplicação vai funcionar normalmente agora!

Se quiser saber mais sobre o Eject do Expo, dê uma olhada na [documentação oficial](#) ;)

## Conclusão

Vimos, nesse artigo, quais são as principais diferenças entre fazer um aplicativo usando Expo e React Native CLI, além de mostrar algumas vantagens e desvantagens de cada um.

Em resumo, se você é iniciante no React Native, começar pelo Expo pode ser muito bom, já que ele te facilita muito na configuração inicial do ambiente e isso ajudará a se familiarizar com essa biblioteca JavaScript.

Mas, se você já é um desenvolvedor ou quer se aprofundar em React Native, usar a versão CLI te fornecerá uma gama muito maior de ferramentas, ou seja, é possível usar todo o potencial dessa biblioteca.

### Confira neste artigo:

- [Tópicos](#)
- [O que é CLI?](#)
- [O que é o Expo?](#)
- [Diferenças entre o Expo CLI e o React Native CLI](#)
- [Migração do projeto Expo para o React Native puro](#)
- [Conclusão](#)

[Artigo Anterior](#)[Próximo Artigo](#)[Como gerenciar estados com Flutter Provider](#)[Configurando o ambiente React Native](#)

Veja outros artigos sobre  
[Mobile](#)

## Quer mergulhar em tecnologia e aprendizagem?

Receba a newsletter que o nosso CEO escreve pessoalmente, com insights do mercado de trabalho, ciência e desenvolvimento de software

Escreva seu email

**ME INSCREVA**

## Nossas redes e apps



Institucional

A Alura

[Sobre nós](#)

[Como Funciona](#)

[Trabalhe conosco](#)

[Todos os cursos](#)

[Para Empresas](#)

[Depoimentos](#)

[Para Escolas](#)

[Instrutores\(as\)](#)

[Política de Privacidade](#)

[Dev em <T>](#)

[Compromisso de Integridade](#)

[Termos de Uso](#)

[Status](#)

## Conteúdos

[Alura Cases](#)

[Imersões](#)

[Artigos](#)

[Podcasts](#)

[Artigos de educação corporativa](#)

## Fale Conosco

[Email e telefone](#)

[Perguntas frequentes](#)

## Novidades e Lançamentos

**ENVIAR**

## CURSOS

### Cursos de Programação

[Lógica](#) | [Python](#) | [PHP](#) | [Java](#) | [.NET](#) | [Node JS](#) | [C](#) | [Computação](#) | [Jogos](#) | [IoT](#)



## **Cursos de Front-end**

HTML, CSS | React | Angular | JavaScript | jQuery

## **Cursos de Data Science**

Ciência de dados | BI | SQL e Banco de Dados | Excel | Machine Learning | NoSQL | Estatística

## **Cursos de DevOps**

AWS | Azure | Docker | Segurança | IaC | Linux

## **Cursos de UX & Design**

Usabilidade e UX | Vídeo e Motion | 3D

## **Cursos de Mobile**

React Native | Flutter | iOS e Swift | Android, Kotlin | Jogos

## **Cursos de Inovação & Gestão**

Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas