

ENTRAR

MATRICULE-SE

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

Configurando o ambiente React Native

**andre-cunha5**

14/09/2021

COMPARTILHE

Configurando o ambiente React Native

Você veio do React e tem interesse em usar seus conhecimentos em um projeto? Quer fazer um app? Ou, talvez, quer construir um aplicativo multiplataforma usando uma poderosa biblioteca Javascript criada pelo Facebook? Então, esse artigo é pra você! Nele mostraremos como configurar o seu [ambiente React Native](https://www.alura.com.br/artigos/configurando-o-ambiente-react-native) nos sistemas Linux,

Windows e macOS. Você aprenderá a como executar as aplicações tanto em emuladores de Android e iOS, e também em dispositivos físicos.

Vamos lá?

Tópicos

Neste artigo, vamos ver:

- Aplicações com React Native Android
 - Windows
 - macOS
 - Linux
 - Usando o Emulador
 - Usando dispositivo Android físico
- Aplicações com React Native iOS
 - macOS

Aplicações com React Native Android

Windows

Para começar, faremos 4 instalações básicas para configurar o ambiente Windows. São elas:

- Chocolatey;
- Node.js na versão LTS;
- JDK 11;
- Android Studio.

Primeiro, vamos instalar o Chocolatey. Ele nos permitirá gerenciar os pacotes do Windows, para podermos instalar ferramentas e dependências pelo terminal. Abra o *powershell* como administrador, usando o comando `Windows + X`. Ou pesquise nos seus aplicativos por *powershell*, clique com o botão direito sobre ele e selecione a opção que tem "Admin".

Verifique suas permissões com o comando:

```
Get-ExecutionPolicy
```

Se aparecer a mensagem "Restricted" (Restrito), execute o comando abaixo, caso contrário, pule para o próximo comando:

```
Set-ExecutionPolicy AllSigned
```

Instalaremos o chocolatey com o seguinte comando:

```
Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager::SecurityProtocol = TLS12]; Invoke-WebRequest https://chocolatey.org/install.ps1 -OutFile $env:temp\chocolatey.ps1
```

Teste se a instalação foi bem sucedida com o comando:

```
choco
```

Agora, feche e abra o seu *powershell*. Para instalar o Node.js e o SDK 11, faremos no *powershell*:

```
choco install -y nodejs-lts openjdk11
```

Novamente, reinicie o *powershell*, fechando e abrindo ele. Rode os comandos abaixo para conferir a instalação:

```
node -v
```

```
java -version
```

Se tudo ocorrer bem, serão exibidas as versões do node e do java instaladas na sua máquina.

E, por fim, instalaremos a versão mais estável do Android Studio. Para isso, é recomendado seguir o passo a passo que está no próprio site do Android Studio, visto que constantemente está sendo atualizado. Entre no primeiro link abaixo para baixar o Android Studio, depois entre no segundo link e navegue até o sistema Windows. Veja que são passos simples e tem até um vídeo demonstrando o que deve ser feito para instalar o Android Studio.

- 1) [Site Oficial para baixar o Android Studio](#);
- 2) [Site Oficial para instalar o Android Studio](#).

Pronto! Depois de completar esses passos, podemos começar a criar nossa aplicação.

macOS

Faremos 5 instalações básicas para configurar o ambiente macOS, são elas:

- Homebrew;
- Node.js;
- Watchman;
- JDK 11;
- Android Studio.

Para instalar o Homebrew, abra o terminal e rode o seguinte comando nele:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/ins
```

Para instalar o Node.js e Watchman, execute no terminal:

```
brew install node@14 watchman
```

Verifique a instalação digitando:

```
node -v
```

Agora, entre nesse [site](#) e baixe o “macOS Installer”. Depois, execute o .dmg e instale o JDK seguindo o passo a passo da instalação.

E, por fim, instalaremos a versão mais estável do Android Studio. Para isso, recomendamos seguir o passo a passo que está no próprio site do Android Studio, visto que constantemente está sendo atualizado. Entre no primeiro link abaixo para baixar o Android Studio, depois entre no segundo link e navegue até o sistema Mac. Veja que são passos simples e tem até um vídeo demonstrando o que deve ser feito para instalar o Android Studio.

3) [Site Oficial para baixar o Android Studio](#).

4) [Site Oficial para instalar o Android Studio](#).

Linux

Faremos 4 instalações básicas para configurar o ambiente Linux. São elas:

- [Curl](#);
- Node.js na versão LTS;
- JDK 11;
- Android Studio.

É importante notar que há vários tipos de ambiente Linux (Ubuntu, Debian etc), e cada versão requer uma instalação adequada. Primeiro, vamos ver o passo a passo para Ubuntu (Debian) e, em seguida, para Arch Linux.

Para Ubuntu (Debian)

Primeiro, vamos instalar o cURL no sistema. Basta abrir o terminal e digitar:

```
sudo apt-get install curl
```

Agora, vamos instalar o Node.js pelas seguintes linhas de comandos:

```
# para Ubuntu
curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
sudo apt-get install -y nodejs

# para Debian
sudo curl -sL https://deb.nodesource.com/setup_14.x | bash -
sudo apt-get install -y nodejs
```

Para verificar se a instalação do node foi feita com sucesso, execute:

```
node --version
```

Será mostrada a versão do node instalada. Aparecerá algo como “v14.xx.x” no terminal.

Agora, instalaremos o JDK 11 (Java Development Kit), usando os seguintes comandos:

```
sudo add-apt-repository ppa:openjdk-r/ppa  
sudo apt-get update  
sudo apt-get install openjdk-11-jdk
```

Para verificar se a instalação do JDK foi feita com sucesso, execute:

```
java --version
```

Caso apareça mais de uma versão do Java instalada na sua máquina, selecione a versão 11 como padrão com o seguinte comando:

```
sudo update-alternatives --config java
```

Para Arch Linux

Começaremos instalando o Node com o seguinte comando:

```
sudo pacman -S nodejs-lts-fermium npm
```

Para verificar se a instalação do node foi feita com sucesso, execute:

```
node --version
```

Agora, instalaremos o JDK 11 (Java Development Kit) usando os seguintes comandos:

```
sudo pacman -S jdk11-openjdk
```

Depois, precisamos setar e definir a versão 11 como padrão:

```
archlinux-java set java-11-openjdk
```

E, por fim, independente se é o sistema Linux Ubuntu Debian ou Arch Linux, instalaremos a versão mais estável do Android Studio. Para isso, é recomendado seguir o passo a passo que está no próprio site do Android Studio, visto que constantemente está sendo atualizado. Entre no primeiro link abaixo para baixar o Android Studio, depois entre no segundo link e navegue até o sistema Linux e veja que são passos simples. Tem até um vídeo demonstrando o que deve ser feito para instalar o Android Studio.

5) [Site Oficial para baixar o Android Studio](#).

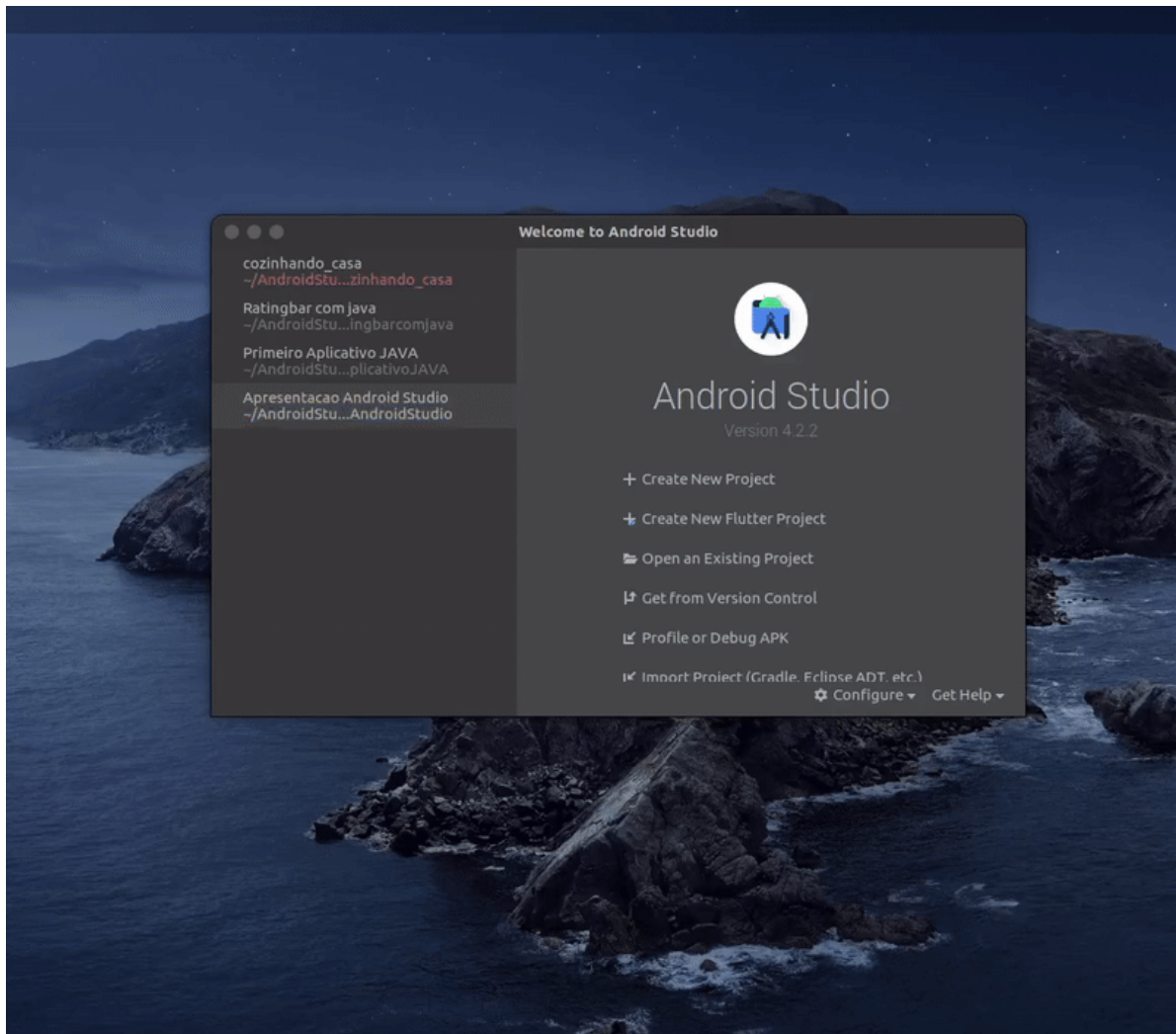
6) [Site Oficial para instalar o Android Studio](#).

Usando o Emulador

Depois de baixarmos o Android Studio, vamos ver passo a passo como executar nosso emulador Android. Lembramos que, se seu computador tiver configurações modestas (por exemplo, ser i3, ter 4gb de RAM), talvez não dê para rodar o emulador ou ele pode travar muito. Nesse último caso, a dica é que você use um dispositivo físico para rodar seu app nele (pode ser o seu celular). Para usar o Emulador, siga o passo a passo abaixo:

1. Execute o Android Studio e, na parte inferior direita do aplicativo, selecione “Configure” e depois clique em “AVD Manager”. Será mostrada uma tela com os emuladores instalados.
2. Talvez já apareça um emulador configurado, se na hora da instalação você tiver selecionado “Android Virtual Device” - nessa situação, aperte no ícone de play (em formato de triângulo) para rodar o emulador. Caso não apareça um emulador, vamos configurar um agora mesmo!
3. Clique em “Create Virtual Device”.
4. Escolha o dispositivo que deseja ter como emulador e clique em “Next” (Próximo).

5. Agora, você precisa escolher a imagem do dispositivo. É bom escolher e baixar a versão recomendada pelo Android Studio, ou uma versão de API mais recente (28 ou superior) que são x86. Depois de baixar, o que pode levar um tempo, selecione a imagem baixada e clique em “Next” (Próximo).
6. Será exibida uma tela de configurações do seu emulador. No campo AVD Name, coloque um nome que preferir para seu emulador.
7. Clique em “Finish” (Finalizar).
8. Seu emulador deve estar sendo exibido na lista. Basta clicar no ícone de play para executá-lo. Lembramos que esse processo pode demorar pela primeira vez.



Vamos, agora, executar o app no emulador Android. Primeiro, será necessário ter criado um projeto [React Native](#). Você já tem um projeto? Caso não tenha feito um ainda, navegue até a pasta do seu computador onde deseja salvá-lo, abra o terminal e digite:

```
npx react-native init nomedoapp
```

Depois, abra dois terminais dentro da pasta do seu projeto React Native. Em nosso exemplo, a pasta tem o nome “nomedoapp”.

No terminal 1, digite:

```
npm start
```

No terminal 2, digite:

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

Dica: Sempre lembre de abrir, primeiro, o emulador e, só depois, rodar a aplicação React Native no terminal. Se não, o processo pode causar algum erro. Ou seja, a gente segue a ordem de:

1. Abrir o emulador;
2. Rodar a aplicação React Native no terminal.

Nada complicado, não é mesmo?

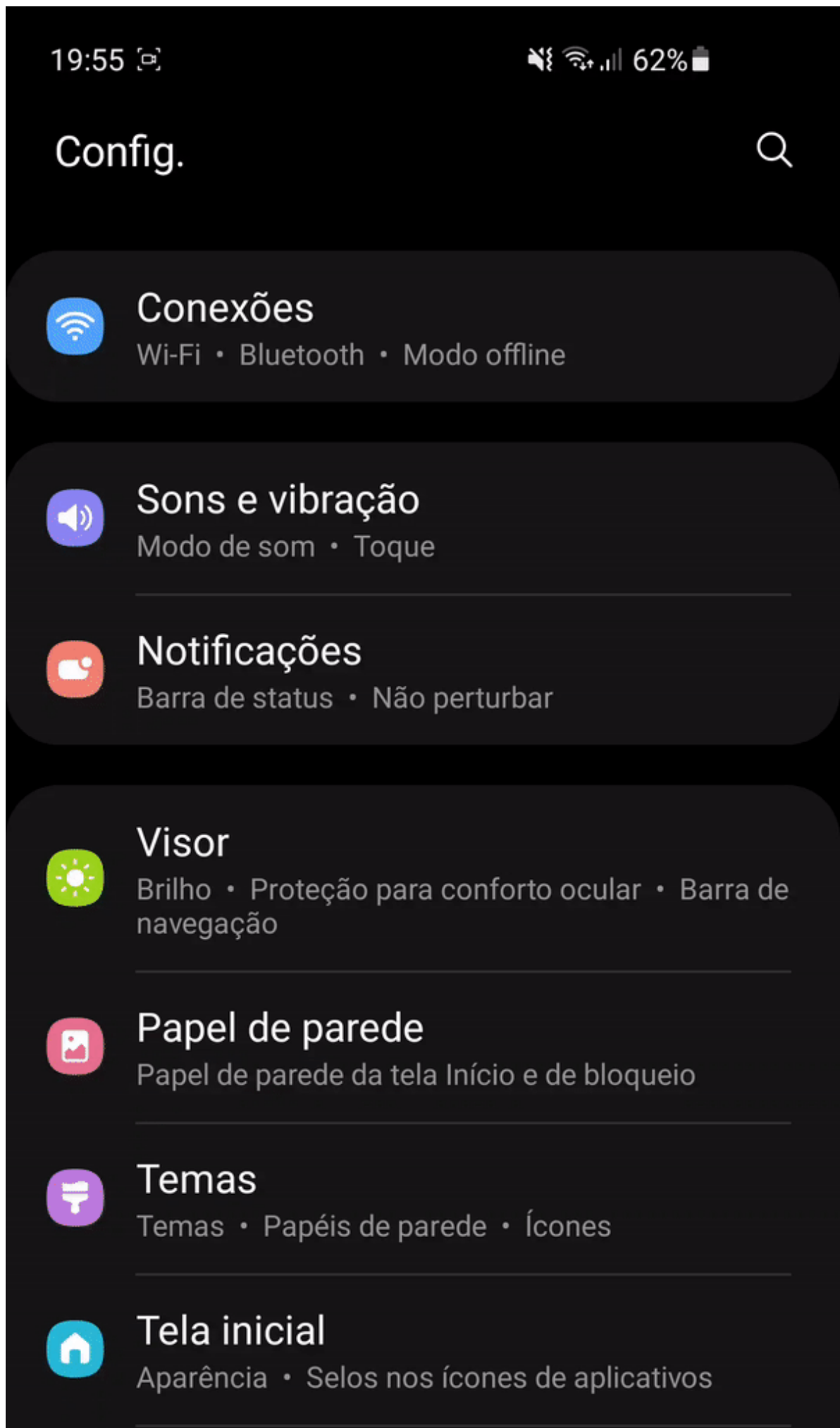
Usando dispositivo Android físico

Se nosso computador possui uma configuração mais básica, ajuda muito usar um celular ou outro dispositivo Android físico, conforme dissemos anteriormente. Como fazer isso?

Conecte seu dispositivo via USB no computador. Ele precisa estar com o modo **Depuração USB** habilitado. Caso ainda não esteja, vamos habilitar a **opção de Desenvolvedor**. Siga o passo a passo a seguir:

Observação: Como existem várias versões e tipos de Android, talvez o passo a passo a seguir não tenha as mesmas opções no seu aparelho. Nesse caso, uma pesquisa rápida no google sobre "Habilitar modo desenvolvedor para o celular X" deve resolver.

1. Entre nas configurações do aparelho.
2. Vá para **Sistema** ou **Sobre o telefone**.
3. Caso sua tela anterior tenha sido **Sistema**, clique na opção Sobre. Agora, se sua tela anterior já era o **Sobre**, siga para o próximo passo.
4. Clique em **Informações do software**.
5. Nessa tela aberta, clique repetidamente na opção **Número da versão** ou em **Número de compilação** até ele te tornar um desenvolvedor.
6. Pronto, agora você tem a opção desenvolvedor ativada.



Ativado o modo desenvolvedor, procure a nova opção que apareceu nas suas configurações do celular. Geralmente ela aparece na tela do **Sistema** ou antes dessa tela, com o nome de **Opções do desenvolvedor**. Entre neste menu e ative a opção **Depuração USB**.

22:31

 35%

< Informações do software

Versão do One UI

3.1

Versão Android

11

Atualização do sistema Google Play

1 de junho de 2021

Versão de banda de base

G973FXXSBFUE6

Versão kernel

4.14.113-21644994

#1 Fri Jun 18 16:26:54 KST 2021

Número de compilação

RP1A.200720.012.G973FXXSBFUF3

Status SE para Android

Enforcing

SEPF_SM-G973F_11_0010

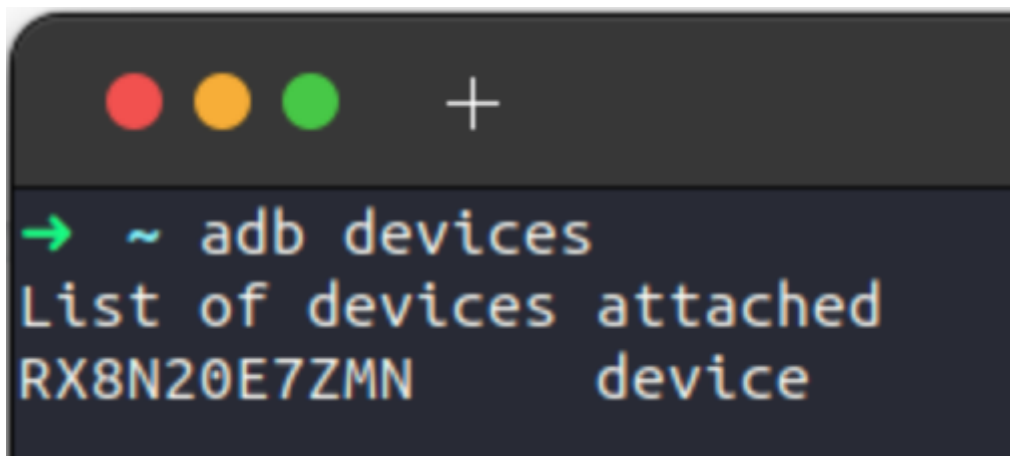
Fri Jun 18 16:40:41 2021

Pronto! Com a opção **Depuração USB** ativada, podemos usar o dispositivo físico!

Agora, com o seu aparelho conectado no cabo USB no computador, abra o terminal e digite:

```
adb devices
```

Deverá ser exibida uma lista com os celulares conectados no computador. Isso significa que estamos prontos para executar a aplicação.



Agora, para executar o aplicativo no seu celular, é necessário que você já tenha criado um projeto React Native. Se não tiver criado ainda, basta executar essa linha de comando no terminal (antes, navegue até a pasta do seu computador que deseje salvar o projeto):

```
npx react-native init nomedoapp
```

Agora, serão necessários dois terminais abertos dentro da pasta do projeto criado. No terminal 1, digite:

```
npm start
```

No terminal 2, digite:

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

Será carregado o aplicativo para o seu celular e você poderá testá-lo enquanto programa.

Aplicações com React Native iOS

Infelizmente, ainda só é possível configurar o ambiente iOS de forma nativa nos ambientes macOS. Se seu desejo é programar para iOS e não possui um macOS, uma opção é alugar uma máquina virtual que rode esse sistema operacional, para poder programar e testar seus apps por lá. Abaixo seguem duas sugestões de máquinas virtuais que podem ser alugadas:

- [MacInCloud](#);
- [Virtual Mac OS X](#).

macOS

Pois bem, vamos configurar nosso macOS para programarmos em React Native no iOS. Faremos as instalações dos seguintes pacotes e ferramentas:

- Homebrew
- Node.js
- Watchman
- CocoaPods
- Xcode

Para instalar o Homebrew, abra o terminal e rode o seguinte comando nele:


```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/ins
```

Para instalar o Node.js e Watchman, execute no terminal:

```
brew install node@14 watchman
```

Verifique a instalação digitando:

```
node --version
```

O CocoaPods nos permite gerenciar as dependências que serão instaladas nos projetos com React Native, evitando erros. Digite o comando abaixo para instalar o CocoaPods:

```
sudo gem install cocoapods
```

E por fim vamos instalar o Xcode. É uma ferramenta da Apple que permite simular vários dispositivos da Apple, desde iPads e iPhones (de várias gerações) até smartwatches. Para isso, abra a App Store no seu macOS e busque por Xcode. E depois instale-o. Também é possível baixar o Xcode pelo [site da Apple](#).

Para usar o simulador é necessário que você já tenha criado um projeto React Native. Se não tiver criado ainda, basta executar essa linha de comando no terminal (antes navegue até a pasta do seu computador que deseje salvar o projeto):

```
npx react-native init nomedoapp
```

Agora serão necessários dois terminais abertos dentro da pasta do projeto criado. No terminal 1 digite:

```
npm start
```

No terminal 2 digite:

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

O simulador iOS vai abrir automaticamente e já vai começar a rodar o seu app nele.

Conclusão

Parabéns se você chegou até aqui! Neste artigo, aprendemos a configurar o React Native nos mais diversos ambientes: Windows, Linux e macOS, tanto para Android quanto para iOS. Aprendemos a configurar os nossos emuladores e até mesmo nossos dispositivos Android físicos para podermos rodar nossos apps enquanto os desenvolvemos.

Agora é com você, o que está esperando para começar a criar seu app? Venha conferir nossos cursos para não perder nem mais um segundo! ;)

Confira neste artigo:

- [Configurando o ambiente React Native](#)
- [Tópicos](#)
- [Aplicações com React Native Android](#)
- [Aplicações com React Native iOS](#)
- [Conclusão](#)

[Artigo Anterior](#)

[Como fazer a migração do Expo para React Native CLI](#)

[Próximo Artigo](#)

[Flutter: o mercado para Dev Jr.](#)

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

Sobre nós

Trabalhe conosco

Para Empresas

A Alura

Como Funciona

Todos os cursos

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