



Preparando o ambiente: rodando o app no simulador



<<

Chegou o momento de configurarmos os nossos simuladores, vamos lá?



1. Aplicações com React Native para Android

1.1. Windows

Para começar, faremos 4 instalações básicas para configurar o ambiente no Windows. São elas:

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

Primeiro, vamos instalar o Chocolatey para gerenciar os pacotes do Windows e 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

COPIAR CÓDIGO



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



Set-ExecutionPolicy AllSigned

COPIAR CÓDIGO



CURSO

Instale o Chocolatey com o seguinte comando:

Set-ExecutionPolicy Bypass -Scope Process -Force;

COPIAR CÓDIGO

Teste se a instalação foi bem-sucedida com o comando:

choco

COPIAR CÓDIGO

Agora, abra o seu *PowerShell*. Para instalar o <u>Node.js</u>
(https://www.alura.com.br/artigos/node-js) e o SDK, execute o comando abaixo no *PowerShell*:

choco install -y nodejs-lts openjdk

COPIAR CÓDIGO

CURSO

Novamente, reinicie o *PowerShell*, fechando e abrindo-o novamente. Rode os comandos abaixo para conferir a instalação:



Se tudo ocorrer bem, serão exibidas as versões do Node.js e do Java instaladas na sua máquina.

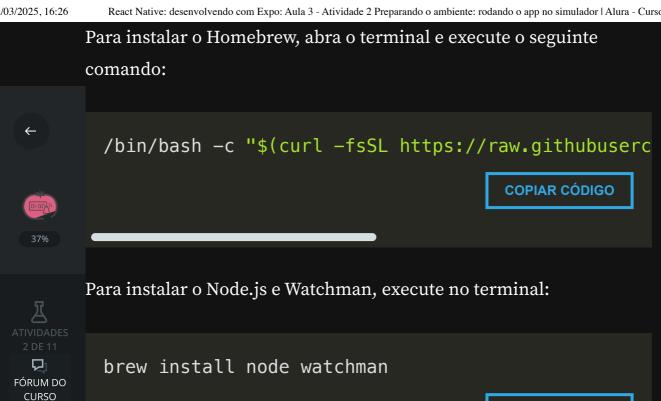
E, por fim, instalaremos a versão mais estável do Android Studio. Para isso, é recomendado seguir o passo a passo que está no próprio site do <u>Android Studio (https://developer.android.com/studio?hl=pt-br)</u>.

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

1.2. macOS

Faremos 5 instalações básicas para configurar o ambiente no macOS, são elas:

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



Verifique a instalação digitando:

node -v

COPIAR CÓDIGO

COPIAR CÓDIGO

Agora, entre neste site

(https://www.oracle.com/java/technologies/downloads/#jdk17-mac) e baixe o "macOS Installer". Depois, execute o arquivo ".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 oficial para baixar o Android Studio (https://developer.android.com/studio?hl=pt-br), visto que está em constante atualização.

1.3. Linux

Faremos 4 instalações básicas para configurar o ambiente no Linux. São elas:



- <u>Curl (https://www.alura.com.br/artigos/curl-como-usar)</u>;
- Node.js na versão LTS;
- DI:OPA
- JDK;
- Android Studio.



Perceba 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.

1.3.1. Para Ubuntu (Debian)

Primeiro, vamos instalar o cURL no sistema. Abra o terminal e digite:

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

COPIAR CÓDIGO

Agora, vamos instalar o Node.js com os seguintes comandos:

```
# para Ubuntu
curl -sL https://deb.nodesource.com/setup_18.x | s
sudo apt-get install -y nodejs

# para Debian
sudo curl -sL https://deb.nodesource.com/setup_22.
sudo apt-get install -y nodejs

COPIAR CÓDIGO
```

Para verificar se a instalação do Node.js foi feita com sucesso, execute:



node --version



COPIAR CÓDIGO



Será mostrada a versão do Node.js instalada.



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

```
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get update
sudo apt-get install openjdk-17_jdk
```

COPIAR CÓDIGO

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

```
java --version
```

COPIAR CÓDIGO

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

sudo update-alternatives --config java

COPIAR CÓDIGO

1.3.2. Para Arch Linux

Começaremos instalando o Node.js com o seguinte comando:



sudo pacman -S nodejs-lts-fermium npm



COPIAR CÓDIGO



Para verificar se a instalação do Node.js foi feita com sucesso, execute:

node --version

COPIAR CÓDIGO

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

sudo pacman -S jdk17-openjdk

COPIAR CÓDIGO

Depois, precisamos configurar e definir a versão mais recente como padrão, no nosso caso, é a 23:

archlinux-java set java-17-openjdk

COPIAR CÓDIGO

E, por fim, independentemente se é o sistema Linux Ubuntu, Debian ou Arch Linux, instalaremos a versão mais estável do Android Studio.

Novamente, recomendamos seguir o passo a passo que está no próprio site do <u>site oficial para baixar o Android Studio</u>
(https://developer.android.com/studio?hl=pt-br).



Usando o emulador





Depois de baixar o Android Studio, vamos ver passo a passo como executar nosso emulador Android.



Lembre-se de que, se seu computador tiver configurações modestas (por exemplo, ser i3, ter 4GB de RAM), talvez não seja possível rodar o emulador ou ele pode travar muito. Nesse caso, a dica é usar um dispositivo físico para rodar seu app (pode ser o seu celular). Para usar o Emulador, siga o passo a passo abaixo:

- Execute o Android Studio e procure a seção "Virtual Device
 Manager", que pode estar na parte inicial do seu Android Studio,
 caso seja sua primeira vez acessando. Caso essa opção não
 apareça na sua tela inicial, você também pode acessá-la clicando
 nos "3 pontos";
- Talvez já apareça um emulador configurado, se na 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!
- Clique em "Create Virtual Device" ("criar dispositivo virtual");
- Escolha o dispositivo que deseja ter como emulador e clique em "Next" ("próximo").
- Agora, você precisa escolher a imagem do dispositivo. É
 recomendável escolher e baixar a versão recomendada pelo
 Android Studio, ou uma versão de API mais recente que seja x86.

Depois de baixar, o que pode levar um tempo, selecione a imagem baixada e clique em "Next" (Próximo).



• Será exibida uma tela de configurações do seu emulador. No campo AVD Name, coloque um nome que preferir para seu emulador.



• Seu emulador deve estar sendo exibido na lista. Basta clicar no ícone de play para executá-lo. Esse processo pode demorar na primeira vez.



Depois de configurar seu emulador, abra dois terminais dentro da pasta do seu projeto React Native.

No terminal 1, digite:

npx react-native run-android

COPIAR CÓDIGO

No terminal 2, digite:

npm **start**

COPIAR CÓDIGO

Dica: Sempre lembre-se de abrir primeiro o emulador e, só depois, rodar a aplicação React Native no terminal. Caso contrário, o processo pode gerar algum erro.

Aplicações com React Native para iOS

Infelizmente, ainda só é possível configurar o ambiente iOS de forma nativa nos ambientes macOS.

Se seu desejo é programar para iOS, mas 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. A seguir, duas sugestões de máquinas virtuais que podem ser alugadas:



- <u>MacInCloud (https://www.macincloud.com/)</u>;
- <u>Virtual Mac OS X (https://virtualmacosx.com/).</u>



macOS

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

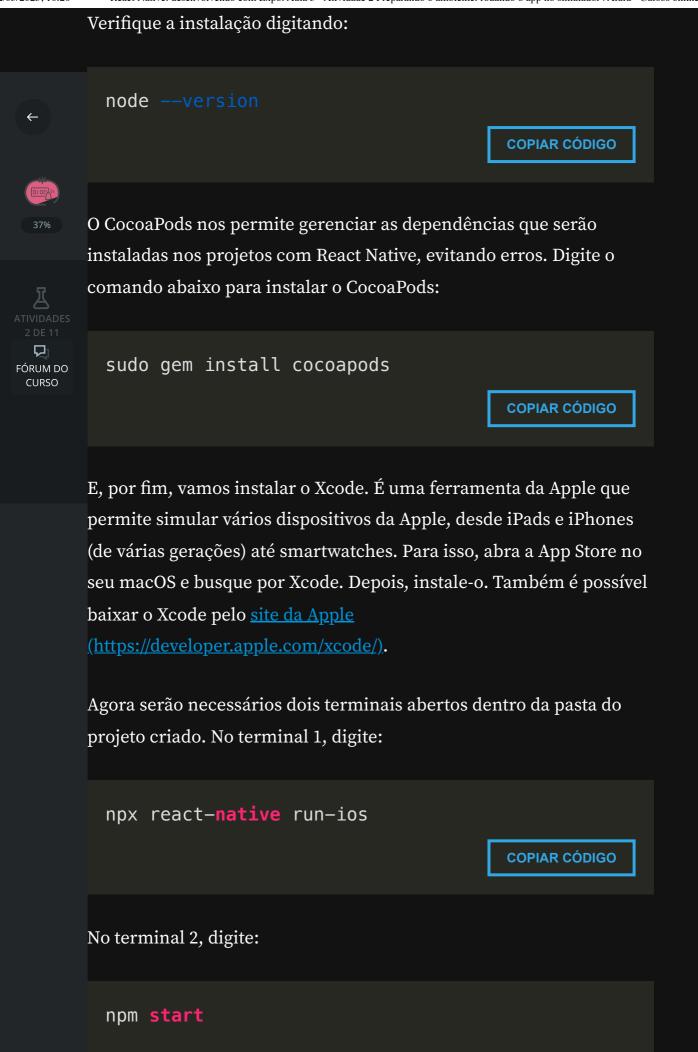
Para instalar o Homebrew, abra o terminal e execute o seguinte comando:

/bin/bash -c "\$(curl -fsSL https://raw.githubuserc

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

brew install node watchman

COPIAR CÓDIGO



COPIAR CÓDIGO



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





Terminou a instalação e preparação do seu ambiente?



FÓRUM DO CURSO Espero você na próxima videoaula!