

Configurando o ambiente de desenvolvimento

Esta página o ajudará a instalar e construir seu primeiro aplicativo React Native.

Se você é novo no desenvolvimento móvel, a maneira mais fácil de começar é com o Expo Go. Expo é um conjunto de ferramentas e serviços construídos em torno do React Native e, embora tenha muitos [recursos](#), o recurso mais relevante para nós no momento é que você pode escrever um aplicativo React Native em minutos. Você só precisará de uma versão recente do Node.js e de um telefone ou emulador. Se quiser experimentar o React Native diretamente no seu navegador antes de instalar qualquer ferramenta, você pode experimentar o [Snack](#).

Se você já está familiarizado com o desenvolvimento móvel, talvez queira usar o React Native CLI. Requer Xcode ou Android Studio para começar. Se você já tiver uma dessas ferramentas instalada, poderá colocá-la em funcionamento em alguns minutos. Se eles não estiverem instalados, você deverá gastar cerca de uma hora instalando-os e configurando-os.

Início rápido da Expo Go

Início rápido do React Native CLI

Siga estas instruções se precisar criar código nativo em seu projeto. Por exemplo, se você estiver integrando o React Native em um aplicativo existente ou se executou o "prebuild" do Expo para gerar o código nativo do seu projeto, você precisará desta seção.

As instruções são um pouco diferentes dependendo do seu sistema operacional de desenvolvimento e se você deseja começar a desenvolver para iOS ou Android. Se você deseja desenvolver para Android e iOS, tudo bem - você pode escolher um para começar, já que a configuração é um pouco diferente.

Development OS

Mac OS

janelas

Linux

Target OS

Instalando dependências

Você precisará do Node, da interface de linha de comando React Native, de um JDK e do Android Studio.

Embora você possa usar qualquer editor de sua escolha para desenvolver seu aplicativo, você precisará instalar o Android Studio para configurar as ferramentas necessárias para criar seu aplicativo React Native para Android.

Nó, JDK

Recomendamos instalar o Node via [Chocolatey](#), um gerenciador de pacotes popular para Windows.

Recomenda-se usar uma versão LTS do Node. Se você quiser alternar entre diferentes versões, talvez queira instalar o Node via [nvm-windows](#), um gerenciador de versões do Node para Windows.

React Native também requer [Java SE Development Kit \(JDK\)](#), que também pode ser instalado usando Chocolatey.

Abra um prompt de comando do administrador (clique com o botão direito em prompt de comando e selecione "Executar como administrador") e execute o seguinte comando:

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

Se você já instalou o Node em seu sistema, certifique-se de que seja o Node 16 ou mais recente. Se você já possui um JDK em seu sistema, recomendamos o JDK11. Você pode encontrar problemas ao usar versões superiores do JDK.

Você pode encontrar opções de instalação adicionais na [página de downloads do Node](#).

Se estiver usando a versão mais recente do Java Development Kit, você precisará alterar a versão Gradle do seu projeto para que ele possa reconhecer o JDK. Você pode fazer isso acessando `{project root folder}\android\gradle\wrapper\gradle-`

`wrapper.properties` e alterando o `distributionUrl` valor para atualizar a versão do Gradle. Você pode conferir [aqui](#) os últimos lançamentos do Gradle .

Ambiente de desenvolvimento Android

Configurar seu ambiente de desenvolvimento pode ser um tanto entediante se você for novo no desenvolvimento Android. Se você já estiver familiarizado com o desenvolvimento Android, talvez seja necessário configurar algumas coisas. Em ambos os casos, certifique-se de seguir cuidadosamente as próximas etapas.

1. Instale o Android Studio

Baixe e instale o [Android Studio](#) . No assistente de instalação do Android Studio, certifique-se de que as caixas ao lado de todos os itens a seguir estejam marcadas:

- `Android SDK`
- `Android SDK Platform`
- `Android Virtual Device`
- Se você ainda não estiver usando o Hyper-V: `Performance (Intel ® HAXM)` ([Veja aqui](#) para AMD ou Hyper-V)

Em seguida, clique em “Avançar” para instalar todos esses componentes.

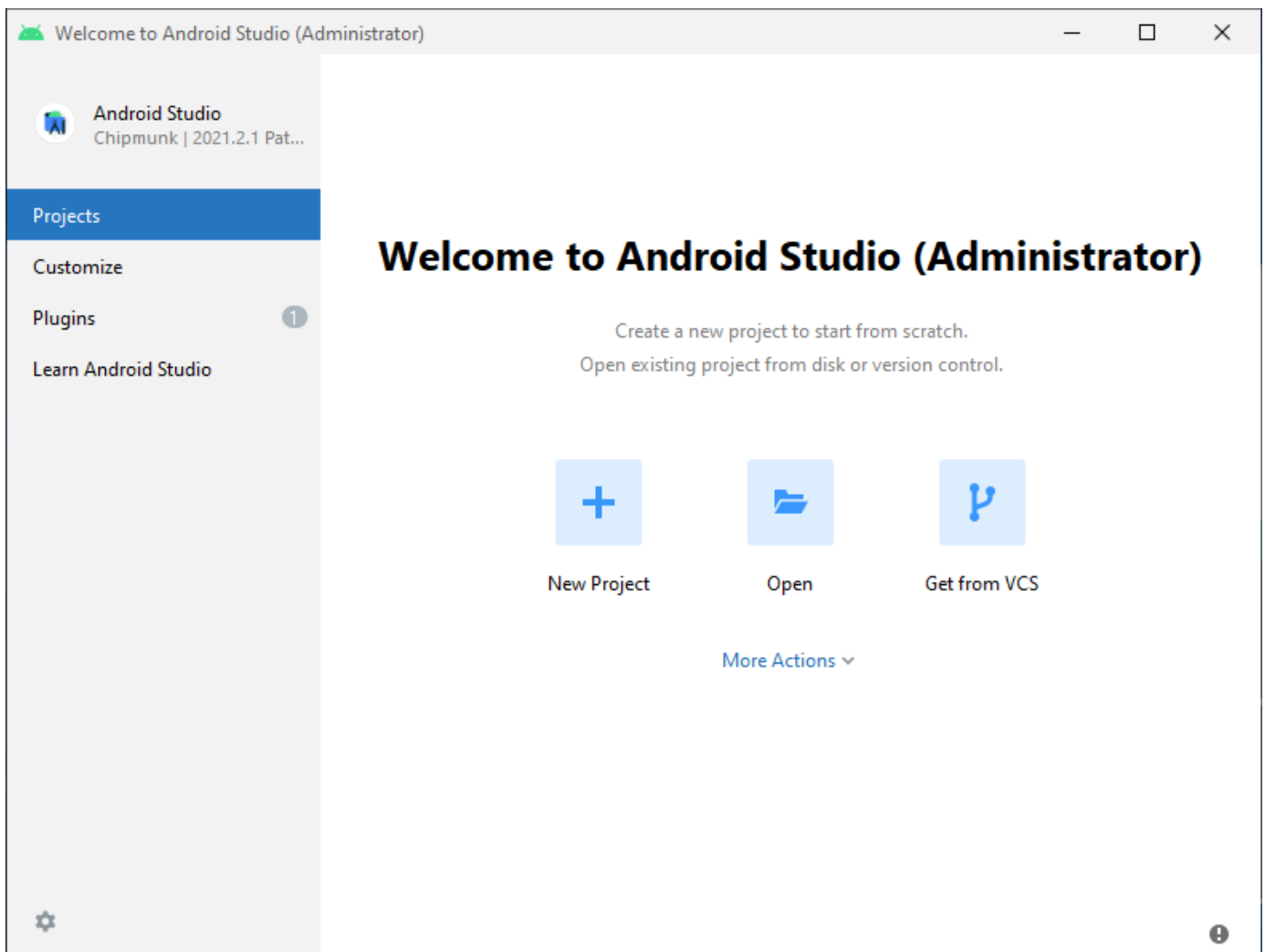
Se as caixas de seleção estiverem esmaecidas, você terá a chance de instalar esses componentes mais tarde.

Depois que a configuração for finalizada e a tela de boas-vindas for exibida, prossiga para a próxima etapa.

2. Instale o Android SDK

O Android Studio instala o Android SDK mais recente por padrão. Construir um aplicativo React Native com código nativo, entretanto, requer o `Android 13 (Tiramisu)` SDK em particular. SDKs Android adicionais podem ser instalados por meio do SDK Manager no Android Studio.

Para fazer isso, abra o Android Studio, clique no botão "Mais ações" e selecione "SDK Manager".



O SDK Manager também pode ser encontrado na caixa de diálogo "Configurações" do Android Studio, em **Idiomas e estruturas** → **Android SDK**.

Selecione a guia "Plataformas SDK" no SDK Manager e marque a caixa ao lado de "Mostrar detalhes do pacote" no canto inferior direito. Procure e expanda a **Android 13 (Tiramisu)** entrada e certifique-se de que os seguintes itens estejam marcados:

- **Android SDK Platform 33**
- **Intel x86 Atom_64 System Image** ou **Google APIs Intel x86 Atom System Image**

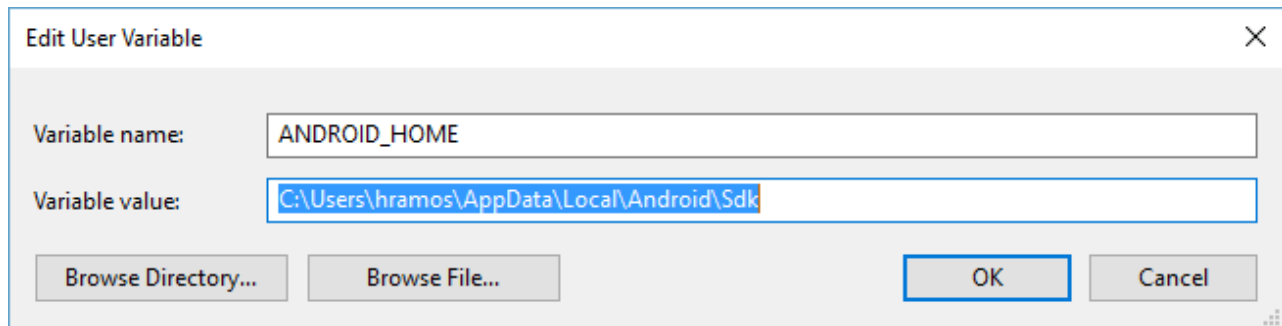
Em seguida, selecione a guia "Ferramentas SDK" e marque a caixa ao lado de "Mostrar detalhes do pacote" aqui também. Procure e expanda a **Android SDK Build-Tools** entrada e certifique-se de que **33.0.0** esteja selecionada.

Por fim, clique em "Aplicar" para baixar e instalar o Android SDK e ferramentas de construção relacionadas.

3. Configure a variável de ambiente **ANDROID_HOME**

As ferramentas React Native requerem a configuração de algumas variáveis de ambiente para construir aplicativos com código nativo.

1. Abra o **Painel de Controle do Windows**.
2. Clique em **Contas de usuário** e clique em **Contas de usuário** novamente
3. Clique em **Alterar minhas variáveis de ambiente**
4. Clique em **Novo...** para criar uma nova `ANDROID_HOME` variável de usuário que aponte para o caminho para seu Android SDK:



O SDK é instalado, por padrão, no seguinte local:

```
%LOCALAPPDATA%\Android\Sdk
```

Você pode encontrar a localização real do SDK na caixa de diálogo "Configurações" do Android Studio, em **Idiomas e estruturas** → **Android SDK**.

Abra uma nova janela do prompt de comando para garantir que a nova variável de ambiente seja carregada antes de prosseguir para a próxima etapa.

1. Abra o PowerShell
2. Copie e cole **Get-ChildItem -Path Env:** no PowerShell
3. Verificar `ANDROID_HOME` foi adicionado

4. Adicione ferramentas de plataforma ao Path

1. Abra o **Painel de Controle do Windows**.
2. Clique em **Contas de usuário** e clique em **Contas de usuário** novamente
3. Clique em **Alterar minhas variáveis de ambiente**
4. Selecione a variável **Caminho**.
5. Clique em **Editar**.

6. Clique em **Novo** e adicione o caminho para ferramentas de plataforma à lista.

O local padrão para esta pasta é:

```
%LOCALAPPDATA%\Android\Sdk\platform-tools
```

Interface de linha de comando nativa do React

React Native possui uma interface de linha de comando integrada. Em vez de instalar e gerenciar uma versão específica da CLI globalmente, recomendamos que você acesse a versão atual em tempo de execução usando o `npx`, que acompanha o Node.js. Com `npx react-native <command>`, a versão estável atual da CLI será baixada e executada no momento em que o comando for executado.

Criando um novo aplicativo

Se você instalou anteriormente um `react-native-cli` pacote global, remova-o, pois pode causar problemas inesperados:

```
npm uninstall -g react-native-cli @react-native-community/cli
```

React Native possui uma interface de linha de comando integrada, que você pode usar para gerar um novo projeto. Você pode acessá-lo sem instalar nada globalmente usando o `npx`, que acompanha o Node.js. Vamos criar um novo projeto React Native chamado "AwesomeProject":

```
npx react-native@latest init AwesomeProject
```

Isso não é necessário se você estiver integrando o React Native em um aplicativo existente, se você "expulsou" da Expo ou se estiver adicionando suporte Android a um projeto React Native existente (consulte [Integração com aplicativos existentes](#)). Você também pode usar uma CLI de terceiros para iniciar seu aplicativo React Native, como [Ignite CLI](#).

[Opcional] Usando uma versão ou modelo específico

Se quiser iniciar um novo projeto com uma versão específica do React Native, você pode usar o `--version` argumento:

```
npx react-native@X.XX.X init AwesomeProject --version X.XX.X
```

Você também pode iniciar um projeto com um modelo React Native personalizado com o `--template` argumento.

Preparando o dispositivo Android

Você precisará de um dispositivo Android para executar seu aplicativo React Native para Android. Pode ser um dispositivo Android físico ou, mais comumente, você pode usar um dispositivo virtual Android que permite emular um dispositivo Android em seu computador.

De qualquer forma, você precisará preparar o dispositivo para executar aplicativos Android para desenvolvimento.

Usando um dispositivo físico

Se você tiver um dispositivo Android físico, poderá usá-lo para desenvolvimento no lugar de um AVD conectando-o ao computador usando um cabo USB e seguindo as instruções [aqui](#).

Usando um dispositivo virtual

Se você usar o Android Studio para abrir o `./AwesomeProject/android`, poderá ver a lista de dispositivos virtuais Android (AVDs) disponíveis abrindo o "AVD Manager" no Android Studio. Procure um ícone parecido com este:



Se você instalou recentemente o Android Studio, provavelmente precisará [criar um novo AVD](#). Selecione "Criar dispositivo virtual...", escolha qualquer telefone da lista e clique em "Avançar" e selecione a imagem **Tiramisu API nível 33**.

Se você não possui o HAXM instalado, clique em "Instalar HAXM" ou siga [estas instruções](#) para configurá-lo e volte para o AVD Manager.

Clique em “Avançar” e depois em “Concluir” para criar seu AVD. Neste ponto, você poderá clicar no botão de triângulo verde próximo ao seu AVD para iniciá-lo e prosseguir para a próxima etapa.

Executando seu aplicativo React Native

Etapa 1: iniciar o Metrô

Primeiro, você precisará iniciar o Metro, o empacotador JavaScript que acompanha o React Native. Metro "recebe um arquivo de entrada e várias opções e retorna um único arquivo JavaScript que inclui todo o seu código e suas dependências." - [Metro Docs](#)

Para iniciar o Metro, execute o seguinte comando dentro da pasta do projeto React Native:

npm Fio

```
npm start
```

Se você está familiarizado com desenvolvimento web, Metro é muito parecido com webpack – para aplicativos React Native. Ao contrário do Kotlin ou do Java, o JavaScript não é compilado – nem o React Native. Agregar não é o mesmo que compilar, mas pode ajudar a melhorar o desempenho de inicialização e traduzir alguns JavaScript específicos da plataforma em JavaScript com suporte mais amplo.

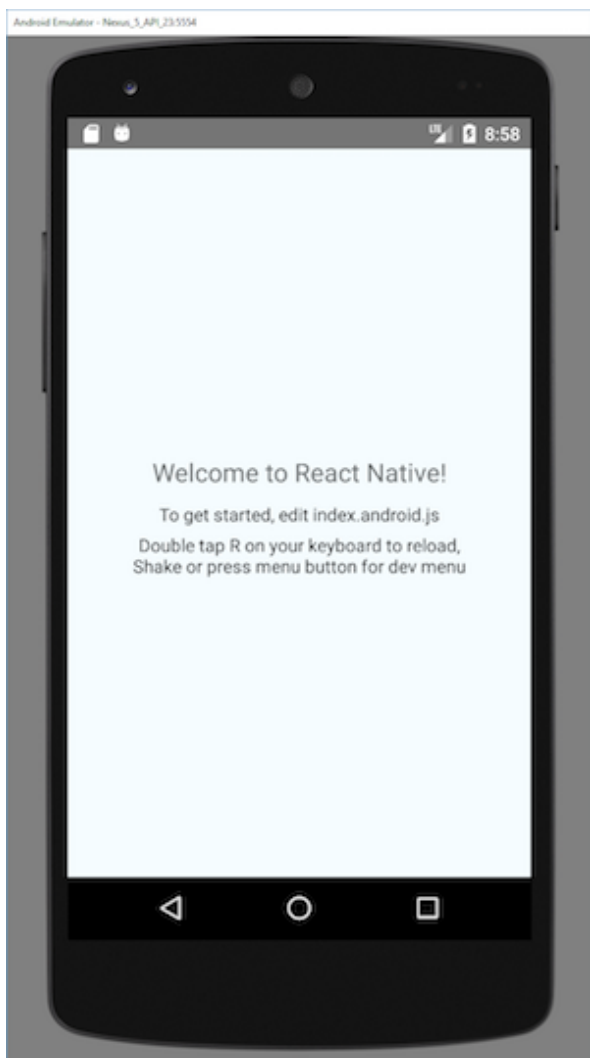
Etapa 2: inicie seu aplicativo

Deixe o Metro Bundler rodar em seu próprio terminal. Abra um novo terminal dentro da pasta do projeto React Native. Execute o seguinte:

npm Fio

```
npm run android
```

Se tudo estiver configurado corretamente, você deverá ver seu novo aplicativo em execução no emulador do Android em breve.



Essa é uma maneira de executar seu aplicativo: você também pode executá-lo diretamente no Android Studio.

Se você não conseguir fazer isso funcionar, consulte a página [Solução de problemas](#).

Modificando seu aplicativo

Agora que você executou o aplicativo com sucesso, vamos modificá-lo.

- Abra `App.tsx` no editor de texto de sua preferência e edite algumas linhas.
- Pressione a `R` tecla duas vezes ou selecione `Reload` no menu Dev (`Ctrl` + `M`) para ver suas alterações!

É isso!

Parabéns! Você executou e modificou com sucesso seu primeiro aplicativo React Native.



O que agora?

- Se você deseja adicionar este novo código React Native a um aplicativo existente, consulte o [Guia de integração](#).

Se você estiver curioso para saber mais sobre o React Native, confira a [Introdução ao React Native](#).

Esta página é útil?  

 [Edite essa página](#)

Última atualização em **29 de agosto de 2023**