

Instalação do React Native

Ivaney Vieira de Sales

08/11/2023

Introdução

O React destaca-se como um renomado framework JavaScript, especialmente relevante para os estudantes de especialização em programação de aplicativos móveis. Sua proeminência reside na aplicação de paradigmas de programação intuitivos, os quais unem o JavaScript de forma coesa a um sistema assemelhado ao HTML denominado JSX. Esta integração possibilita que os desenvolvedores acelerem o processo de desenvolvimento de aplicativos.

O React Native, por sua vez, é um framework que delinea uma hierarquia de código JavaScript composta por componentes de interface do usuário. Ele dispõe de um conjunto específico de componentes para a construção de aplicativos móveis com uma aparência autêntica e funcionalidade adequada tanto para plataformas iOS quanto Android. Apesar das variações de interface, React Native e ReactJS compartilham princípios de design comuns, ambos concebidos pelo Facebook.

A relevância do React Native para estudantes dessa especialização é notável pelos seguintes aspectos:

- Possibilidade de criar interfaces de usuário para plataformas iOS e Android.
- Componentes do React Native têm direitos equivalentes, permitindo a reutilização em aplicativos tanto para Android quanto iOS.
- Desenvolvimento com React Native é notadamente simplificado, rápido e eficiente.

Este guia fornecerá instruções passo a passo, apresentando os requisitos e procedimentos necessários para garantir uma instalação bem-sucedida. Siga cuidadosamente as etapas delineadas abaixo para configurar seu ambiente de programação, permitindo que você comece a desenvolver aplicativos móveis poderosos utilizando a versatilidade e eficácia do React Native no ambiente Linux com Ubuntu.

Atualização do Ubuntu

Para abrir o terminal do Ubuntu 21.10, pressione `Ctrl + Alt + T` no teclado. Você também pode clicar no ícone do terminal na barra lateral do Ubuntu.

O primeiro passo é atualizar o sistema operacional Ubuntu. Para fazer isso, abra o terminal

e execute o seguinte passos:

Atualize a lista de pacotes disponíveis no sistema

```
sudo apt update -y
```

O comando `sudo apt update -y` atualiza a lista de pacotes disponíveis no sistema de forma automatizada, sem exigir intervenção do usuário para confirmar a atualização. Isso é frequentemente usado antes de instalar novos pacotes ou realizar atualizações no sistema para garantir que você esteja usando as versões mais recentes disponíveis.

Se tudo der certo será produzido a seguinte saída no terminal:

```
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Atualize os pacotes instalados no sistema

```
sudo apt upgrade -y
```

O comando `sudo apt upgrade -y`, por sua vez, instala as versões mais recentes dos pacotes instalados no sistema.

O `sudo` é usado para executar o comando com privilégios de superusuário (root), e será possivelmente solicitado que o usuário digite a senha de super usuário. Ele será usado vários comando usado neste tutorial com a mesma finalidade.

O `-y` no final do comando indica que o usuário não será solicitado a confirmar a instalação. Isso é útil ao automatizar a instalação de pacotes, pois não exige que o usuário digite “y” ou “yes” para confirmar a instalação.

Para testar se o sistema está atualizado, execute o seguinte comando:

```
sudo apt list --upgradable
```

Este comando deverá listar os pacotes que podem ser atualizados, se houver algum.

Instalando dependências

Agora que o seu sistema está atualizado, você pode instalar as dependências necessárias para o React Native.

Dependências essenciais

```
sudo apt install -y build-essential libssl-dev libcurl4-openssl-dev libexpat1-dev gettext
```

Esse comando é usado para instalar um conjunto de pacotes essenciais e dependências no sistema. Aqui está uma explicação do que cada parte do comando faz:

A lista de pacotes a serem instalados:

- `build-essential`: Este pacote contém ferramentas e bibliotecas necessárias para compilar programas. Inclui, por exemplo, o compilador GCC.
- `libssl-dev`: Fornece o desenvolvimento de bibliotecas SSL, usadas para criptografia segura.
- `libcurl4-openssl-dev`: Fornece o desenvolvimento de bibliotecas para a biblioteca de transferência de dados com URL, o cURL.
- `libexpat1-dev`: Fornece o desenvolvimento de bibliotecas para o processador XML expat.
- `gettext`: Fornece ferramentas e bibliotecas para internacionalização (i18n) e localização (l10n) de software.
- `unzip`: Ferramenta para descompactar arquivos no formato ZIP.
- `nano`: Editor de texto simples para o terminal.
- `git`: Sistema de controle de versão distribuído.
- `libpulse-dev`: Fornece o desenvolvimento de bibliotecas para o servidor de som PulseAudio.

Esses pacotes são frequentemente necessários para compilar e instalar software a partir do código-fonte, especialmente se o software depende de bibliotecas específicas, como SSL, cURL, ou se envolve o processo de compilação. O comando é útil ao configurar um ambiente de desenvolvimento ou ao instalar software que requer essas dependências.

Kit de desenvolvimento Java

~~O React Native recomenda atualmente a versão 11 do Java SE Development Kit (JDK). Você pode encontrar problemas ao usar versões JDK mais altas. Você pode baixar e instalar o OpenJDK do AdoptOpenJDK ou do seu empacotador de sistema.~~

O seguinte comando instala o OpenJDK 17 no sistema:

```
sudo apt install -y openjdk-17-jdk
```

Para saber se a instalação foi bem sucedida, execute o seguinte comando:

```
java -version
```

A saída do comando `java -version` é mostrada abaixo:

```
openjdk version "17.0.8.1" 2023-08-24
OpenJDK Runtime Environment (build 17.0.8.1+1-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.8.1+1-Ubuntu-0ubuntu122.04, mixed mode, sharing)
```

Aqui, identificamos que a versão do OpenJDK instalada é a 17. No entanto, se uma versão diferente da desejada for exibida, pode indicar a presença de mais de uma versão no sistema. Para confirmar isso, utilize o seguinte comando:

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

A saída do comando `sudo update-alternatives --config java` é mostrada abaixo:

There are 2 choices for the alternative java (providing /usr/bin/java).

Selection	Path	Priority	Status
0	/usr/lib/jvm/java-17-openjdk-amd64/bin/java	1711	auto mode
* 1	/usr/lib/jvm/java-11-openjdk-amd64/bin/java	1111	manual mode
2	/usr/lib/jvm/java-17-openjdk-amd64/bin/java	1711	manual mode

Press <enter> to keep the current choice[*], or type selection number:

Este comando indica que a versão atual do Java é a 11, enquanto a versão 17 também está instalada no sistema. Para escolher a versão 17, digite o número 0 e pressione "Enter".

Android Studio

O Android Studio é um ambiente de desenvolvimento integrado (IDE) para desenvolvimento de aplicativos Android. Ele fornece uma interface gráfica do usuário (GUI) para criar e gerenciar projetos, bem como ferramentas de desenvolvimento para compilar, depurar e testar aplicativos. O Android Studio é baseado no IntelliJ IDEA, um IDE de código aberto para desenvolvimento de software.

1. Crie um diretório Development Tools em seu diretório inicial, chamado DevTools (ou use um nome apropriado):

```
mkdir ~/DevTools
```

Use esse diretório para instalar quaisquer outras ferramentas de desenvolvimento, kits de desenvolvimento de software e outros.

Crie um subdiretório no DevTools chamado Android:

```
mkdir ~/DevTools/Android
```

// Work in progress

```
https://www.linuxfordevices.com/tutorials/linux/installing-android-studio-on-linux
```

```
wget -c https://redirector.gvt1.com/edgedl/android/studio/ide-zips/2022.3.1.21/android-studio-2022.3.1.21-linux.tar.gz
```

```
cd ~/DevTools/Android
```

```
tar -xvzf ~/Downloads/android-studio-2022.3.1.21-linux.tar.gz
```

```
cd android-studio/bin
```

```
./studio.sh
```

Intalando o Android SDK sem o Android Studio

Para obter o Android SDK, o Android Studio é necessário por padrão. Porém, o Android Studio tem 772 MB e ainda faz o download das demais ferramentas necessárias.

Para aqueles que desejam ter apenas o Visual Studio Code como IDE de desenvolvimento móvel, podem obter apenas as ferramentas de comando do Android.

Baixe as ferramentas de comando do Android

No endereço <https://developer.android.com/studio#command-tools>, baixe o arquivo ZIP das ferramentas de comando do Android para o sistema operacional linux.

Somente ferramentas de linha de comando

Plataforma	Pacote de ferramentas do SDK	Tamanho	Soma de verificação SHA-256
Windows	commandlinetools-win-10406996_latest.zip	148,8 MB	9b782a54d246ba5d207110fddd1a35a91087a8aaf4057e9df697b1cbc0ef60fc
Mac	commandlinetools-mac-10406996_latest.zip	148,8 MB	6821609e885d4b68f4066751949a9211f4196ab36df9d63f7a5f9037ca64e2d6
Linux	commandlinetools-linux-10406996_latest.zip	148,8 MB	8919e8752979db73d8321e9babe2caedcc393750817c1a5f56c128ec442fb540

As ferramentas de linha de comando estão incluídas no Android Studio. Se você não precisa do Android Studio, pode fazer o download das ferramentas básicas de linha de comando do Android acima. Você pode usar o [sdkmanager](#) incluído para fazer o download de outros pacotes do SDK.

Figura 1: Download da ferramenta de linha de comando

Desça até o final da página e encontre a seção chamada “Somente Ferramentas de Linha de Comando”.

Procure pelo diretório onde o arquivo foi baixado, normalmente localizado na pasta “Downloads” dentro do diretório principal do usuário.

Crie outro subdiretório, no Android, chamado cmdline-tools

```
mkdir ~/DevTools/Android/cmdline-tools
```

Extraia o arquivo “commandlinetools-linux-*.zip” em um diretório específico:**
~/DevTools/Android

```
unzip ~/Downloads/commandlinetools-linux-*.zip -d ~/DevTools/Android/cmdline-tools  
mv ~/DevTools/Android/cmdline-tools/cmdline-tools ~/DevTools/Android/cmdline-tools/tools
```

No diretório ~/DevTools/Android/cmdline-tools/ também estará os outros subdiretórios do Android SDK. Já que commandlinetools é apenas uma ferramenta Android, assim como as demais que estarão paralelas, tais como: /* Citar outros exemplos */.

Defina o caminho do diretório Android SDK nas variáveis de ambiente do Ubuntu. Edite o seguinte arquivo

```
nano ~/.bashrc
```

Adicione ao final do arquivo .bashrc:

```
export ANDROID_HOME=$HOME/DevTools/Android  
export PATH=$PATH:$ANDROID_HOME/emulator  
export PATH=$PATH:$ANDROID_HOME/cmdline-tools/tools/bin  
export ANDROID_SDK_ROOT=$ANDROID_HOME
```

Recarregue o profile do Linux

```
source ~/.bashrc
```

Quando você faz alterações no seu arquivo .bashrc, como adicionar novas variáveis de ambiente ou definir alias, essas mudanças não têm efeito imediato no terminal atual. Executar o comando este comando força o Bash a ler novamente e aplicar as configurações no terminal atual, sem precisar fechar e abrir um novo terminal.

Testando o sdkmanager

```
sdkmanager --version
```

Ao executar esse comando, a saída geralmente incluirá a versão atual do SDK Manager instalada no seu sistema. Isso pode ser útil para garantir que você esteja usando a versão mais recente ou para fins de referência ao documentar ou depurar problemas relacionados ao ambiente de desenvolvimento Android.

Listando todos os pacotes disponíveis para instalação

```
sdkmanager --list
```

O comando irá listar todos os pacotes SDK do Android disponíveis para instalação. A lista incluirá o nome de cada pacote, sua versão, seu tamanho e uma breve descrição.

Instalando o Android SDK Platform 31

```
sdkmanager "platforms;android-31"
```

Este comando irá baixar e instalar a plataforma SDK Android para Android 31. Uma vez que a instalação for concluída, você poderá usar o SDK para desenvolver aplicativos que são direcionados a dispositivos Android 31.

Aceitando as licenças

```
sdkmanager --licenses
```

A execução deste comando é seguida por uma série de prompts interativos nos quais você pode revisar as licenças e, se concordar com os termos, aceitá-las. Isso é útil em ambientes automatizados ou em scripts de instalação, onde é necessário aceitar as licenças automaticamente para que a instalação prossiga sem intervenção manual.

Lembre-se de que a aceitação de licenças é uma etapa importante no processo de configuração do ambiente de desenvolvimento Android e é necessária para garantir conformidade com os termos de uso associados aos componentes do SDK.

Atualizando os pacotes Android quando necessário

```
sdkmanager --update
```

Instalação dos pacotes necessários:

```
sdkmanager "platform-tools" "build-tools;31.0.0" "extras;android;m2repository" "extras;g  
sdkmanager 'build-tools;34.0.0' emulator platform-tools 'platforms;android-34' 'system-i
```

Ao executar esse comando, você está garantindo que o seu ambiente de desenvolvimento Android está atualizado com as versões mais recentes dos componentes do SDK, o que é importante para garantir a compatibilidade com as versões mais recentes do Android, receber correções de bugs e obter as últimas funcionalidades.

Instalando o Node.js

O Node.js é uma plataforma de software de código aberto que permite que os desenvolvedores criem aplicativos de rede e executem JavaScript fora do navegador. Ele é executado em vários sistemas operacionais, incluindo Linux, macOS e Windows. O Node.js fornece um ambiente de tempo de execução JavaScript que permite aos desenvolvedores criar aplicativos de rede escaláveis. Ele também fornece um rico conjunto de bibliotecas de JavaScript que simplificam o desenvolvimento de aplicativos da web e móveis.

Baixe e importe a chave GPG (Pretty Good Privacy) da Nodsource

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor
```

Crie um repositório DEB para o Node.js:

```
NODE_MAJOR=20
echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_$
```

A variável `NODE_MAJOR` pode ser mudada dependendo da versão do Node.js que você deseja instalar.

NODE_MAJOR	Observação
16	Versão atualmente não suportada
18	Versão atualmente não suportada
20	Versão LTS (Recomendada)
21	Ultima atualização

Onde a versão 20 é atualmente a versão LTS (Long Term Support) do Node.js, ou seja, a versão com suporte a longo prazo.

Execute Atualização e Instalação

```
sudo apt-get update
sudo apt-get install nodejs -y
```

Para testar se o Node.js foi instalado corretamente, execute o seguinte comando:

```
node -v
```

Ao digitar `node -v` e pressionar “Enter” no terminal ou prompt de comando, você receberá como saída a versão específica do Node.js que está instalada no seu computador. Isso serve para verificar rapidamente qual versão do Node.js está em uso, especialmente ao lidar com projetos que podem ter requisitos específicos de versão.

Configurando o Emulador Android

<https://brunorozendo.com/post/criar-avd-gnu-linux.html>

Instalado o React Native CLI

A próxima etapa é instalar o React Native. Deve-se observar que você precisa usar `sudo` para ter o React Native instalado como root, já que iremos instalá-lo globalmente. Este é o código que você deve usar:


```
sudo npm install -g react-native-cli
```

Após a instalação, você pode verificar se o React Native foi instalado corretamente, executando o seguinte comando:

```
react-native --version
```

Primeiro aplicativo React Native