



Configurando ambientes

(Atenção): Você pode configurar o ambiente seguindo as video aulas. Este arquivo é caso queira fazer via PDF.

Windows

iOS

Infelizmente ainda não é possível configurar ambiente de desenvolvimento iOS no Windows, por isso, passe para o ambiente Android.

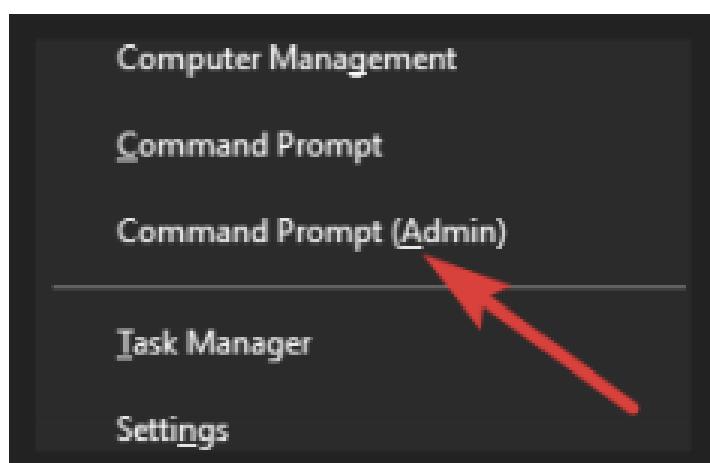
Android

Para configurar o ambiente Android no Windows, vamos precisar instalar 4 dependências: Node, Python2, JDK e Android Studio.

Vamos lá..

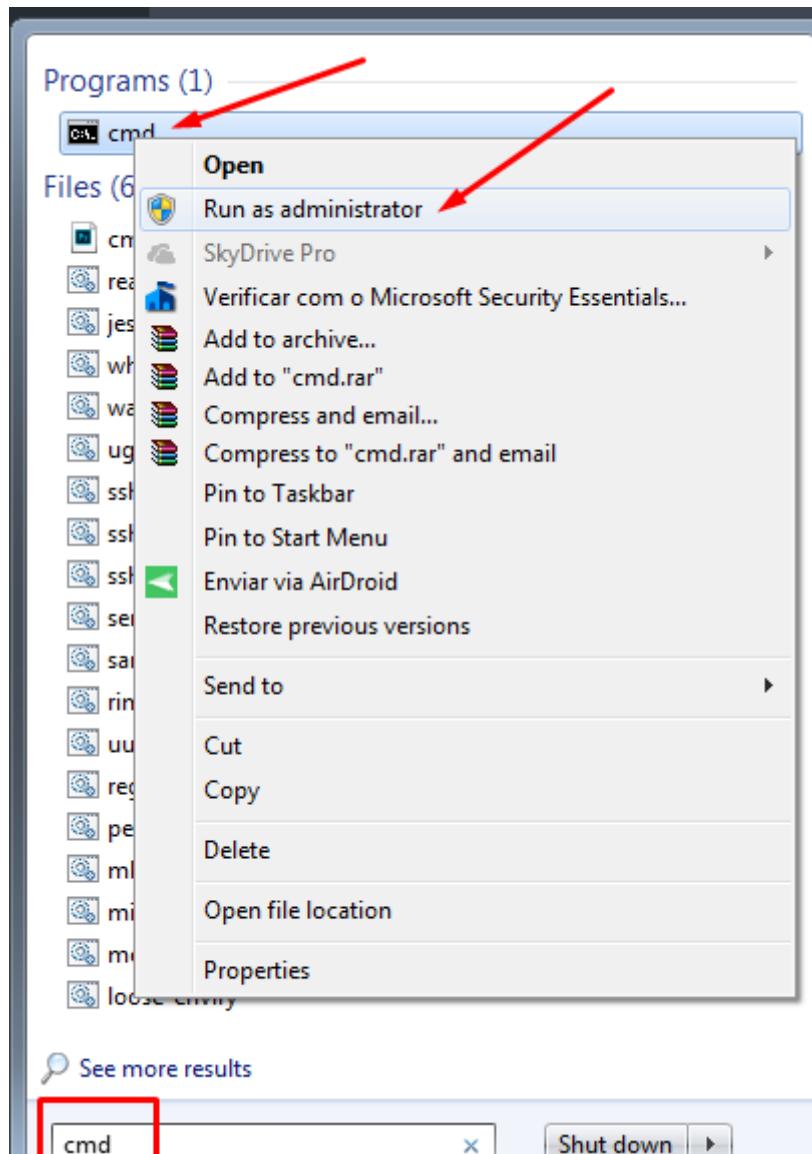
Instalando Chocolatey

Para instalar as libs no Windows, vamos utilizar um Package Manager chamado Chocolatey. Primeiro execute o prompt de comando como administrador clicando com o botão direito sobre o botão “Iniciar”:



Caso esteja no Windows 7:

Aperte o botão iniciar e digite cmd e clique com botão direito sobre o cmd e abra como administrador.



Agora, copie o código contido dentro do link em sua linha de comando de acordo com o prompt que estiver utilizando (se você não sabe qual é a sua opção, provavelmente é a **cmd.exe**):

<http://sujeitoprogramador.com/codigochocolatey.txt>

PowerShell.exe

Antes de executar o comando de instalação, precisamos verificar se você está habilitado para instalar bibliotecas no PowerShell. Dentro do seu terminal, execute o seguinte comando:

[Get-ExecutionPolicy](#)

Se o retorno desse comando for “Restricted”, execute o próximo comando em seu terminal, se não, prossiga para o próximo passo:

[Set-ExecutionPolicy AllSigned](#)

Agora sim, com a devida permissão, execute o próximo comando dentro do link abaixo:

<http://sujeitoprogramador.com/codigopowershell.txt>

Agora, teste se a instalação ocorreu corretamente executando o seguinte comando no seu terminal:

[choco -v](#)

Caso ele devolva a versão do seu chocolatey esta tudo certo e instalado perfeitamente.

Com o Chocolatey instalado, vamos instalar o **NodeJS**, **Python2** e a **JDK8**:

```
choco install -y nodejs.install python2 jdk8
```

Apos carregar e instalar tudo rode o comando para instalar o React Native Cli

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

Se tudo ocorreu bem até aqui, você conseguirá executar o seguinte comando:

```
react-native -h
```

Android Studio

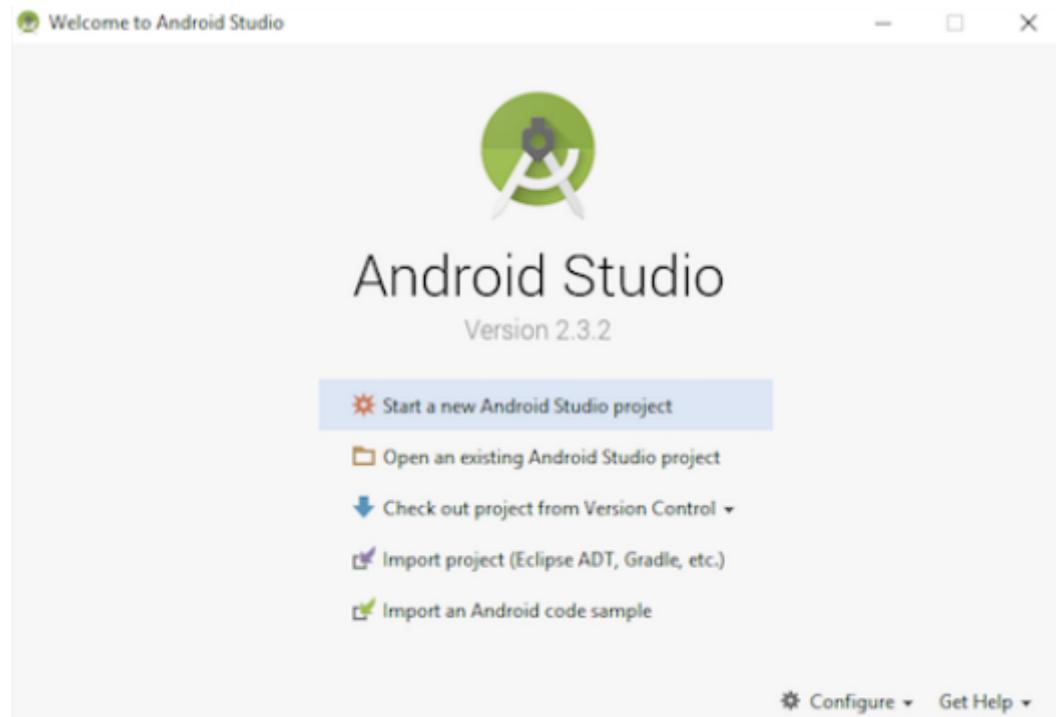
Para emular nossa aplicação vamos precisar de um sistema Android, você pode optar por utilizar seu próprio smartphone ou configurar um emulador no Windows, caso opte por utilizar o próprio dispositivo físico, basta conectá-lo via porta USB e pular o próximo passo. Comece instalando o Android Studio através do link:

<https://developer.android.com/studio/index.html>

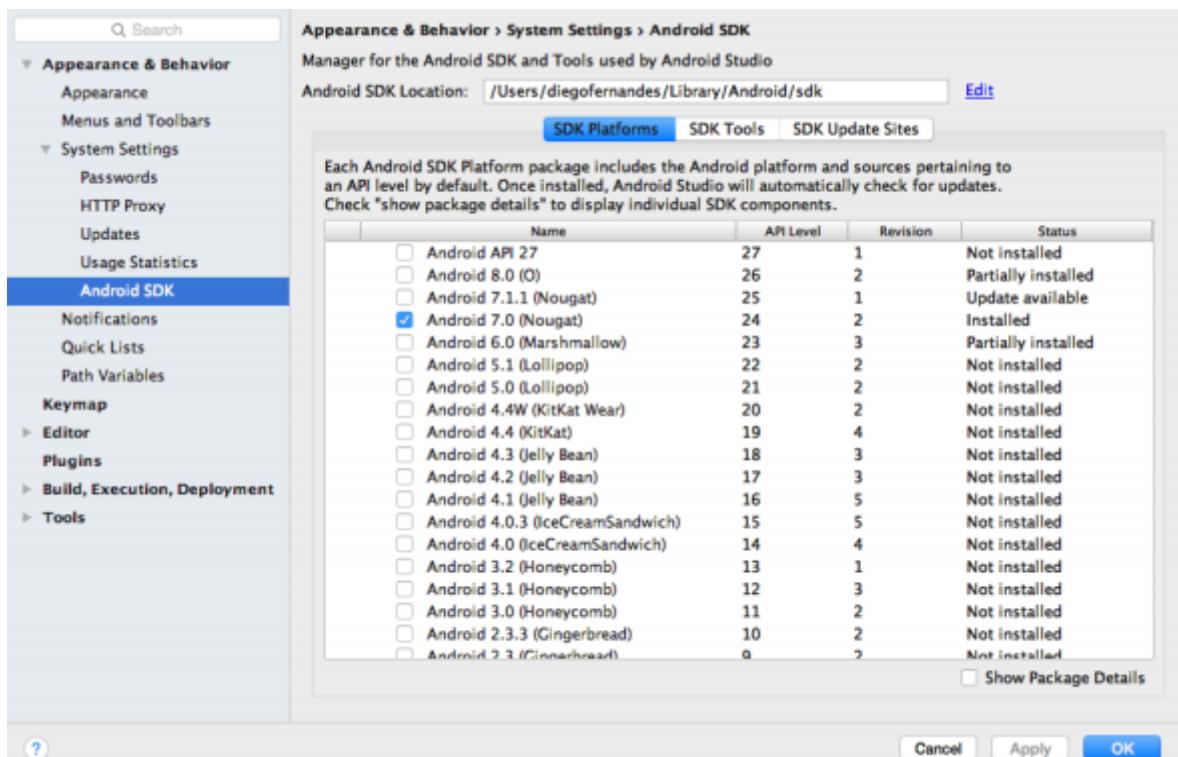
Assim que feito o Download, execute o instalador. No passo “Tipo de instalação” selecione “Custom” ou “Personalizada” e selecione os seguintes componentes:

- Android SDK
- Android SDK Platform
- Performance (Intel ® HAXM)
- Android Virtual Device

Ok se a instalação ocorrer bem, você receberá uma tela como a seguir:



Agora, nessa tela, clique no botão “Configure” e selecione a opção “SDK Manager”. Uma tela como a seguinte deve abrir:



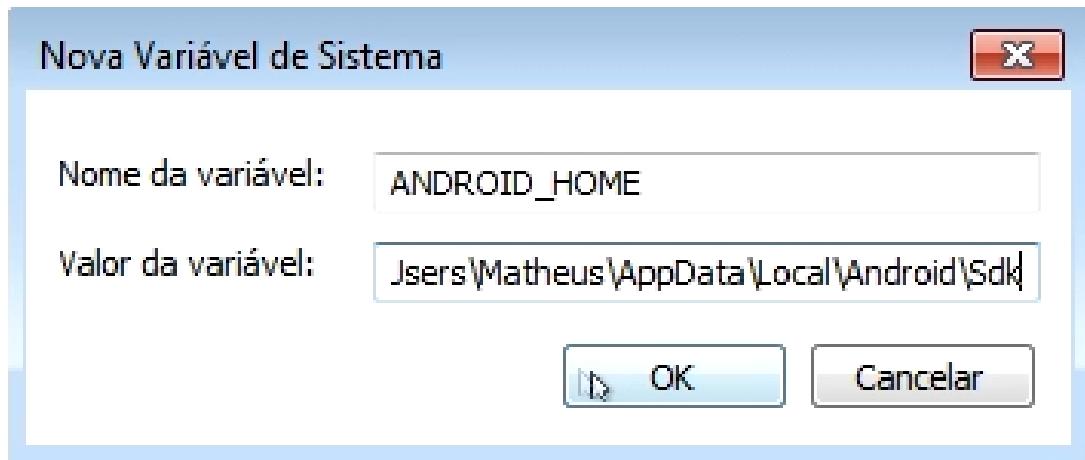
Certifique-se que nessa tela você tenha instalado alguma API superior à 6.0 (no meu caso a 7.0 está instalada). Na aba “SDK Tools”, certifique-se que as seguintes opções estão instaladas. Caso alguma delas ainda não esteja instalada, selecione-a e clique no botão “Apply” no final da tela para realizar o download.

	Name	Version	Status
-	Android SDK Build-Tools 27		Update Available: 27.0.0
<input type="checkbox"/>	GPU Debugging tools		Not Installed
<input type="checkbox"/>	CMake		Not Installed
<input checked="" type="checkbox"/>	LLDB		Not Installed
<input type="checkbox"/>	Android Auto API Simulators	1	Not installed
<input type="checkbox"/>	Android Auto Desktop Head Unit emulator	1.1	Not installed
<input checked="" type="checkbox"/>	Android Emulator	26.1.4	Installed
<input type="checkbox"/>	Android SDK Platform-Tools	26.0.0	Update Available: 26.0.2
<input type="checkbox"/>	Android SDK Tools	26.0.2	Update Available: 26.1.1
<input type="checkbox"/>	Documentation for Android SDK	1	Not installed
<input type="checkbox"/>	Google Play APK Expansion library	1	Not installed
<input type="checkbox"/>	Google Play Licensing Library	1	Not installed
<input type="checkbox"/>	Google Play services	46	Not installed
<input type="checkbox"/>	Google Web Driver	2	Not installed
<input type="checkbox"/>	Instant Apps Development SDK	1.1.0	Not installed
<input type="checkbox"/>	Intel x86 Emulator Accelerator (HAXM installer)	6.2.1	Not installed
<input type="checkbox"/>	NDK	15.2.4203891	Not installed
▼	Support Repository		
<input checked="" type="checkbox"/>	ConstraintLayout for Android		Installed
<input checked="" type="checkbox"/>	Solver for ConstraintLayout		Installed
<input checked="" type="checkbox"/>	Android Support Repository	47.0.0	Installed
<input checked="" type="checkbox"/>	Google Repository	58	Installed

- Vamos configurar a **ANDROID_HOME**

Agora que temos nossa SDK instalada, precisamos configurar a variável ANDROID_HOME em nosso ambiente. Para isso, no SDK Manager do Android Studio, copie o caminho da SDK do Android como na imagem:

Agora, no Painel de Controle do Windows, abra o item “Sistema e Segurança” ou “Sistema”, clique em “Configurações avançadas do sistema”, selecione **“Variáveis de ambiente”** e clique no botão **“Nova variável de ambiente”**, indique o nome da variável como **ANDROID_HOME**, adicione o caminho copiado do Android Studio como segundo parâmetro e clique em OK.



Agora com o ambiente configurado precisamos instalar o emulador. Então agora vá para aula **Aula 6 - Configurando Simulador.**

Linux

iOS

Infelizmente ainda não é possível configurar ambiente de desenvolvimento iOS no Windows, por isso, passe para o ambiente Android.

Android

Para configurar o ambiente Android no Windows, vamos precisar instalar: Node, JDK e Android Studio.

Vamos lá..

Certifique-se que você tenha o CURL instalado executando o seguinte comando no terminal:

`sudo apt-get install curl`

Agora com o CURL instalado, vamos instalar no NodeJS utilizando os seguintes comandos:

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E  
bash -  
sudo apt-get install -y nodejs
```

Agora com o NodeJS instalado, podemos instalar o CLI do React Native:

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

Agora vamos instalar algumas bibliotecas adicionais para instalar o JDK8:

```
sudo apt-get install lib32z1 lib32ncurses5 lib32bz2-1.0  
lib32stdc++6
```

Agora vamos instalar o JDK8 utilizando o comando:

```
sudo apt-get install default-jdk
```

Android Studio

Para emular nossa aplicação vamos precisar do Android Studio, você pode optar por utilizar seu próprio smartphone ou configurar um emulador no Linux, caso opte por utilizar o próprio dispositivo físico, basta conectá-lo via porta USB e pular o próximo passo.

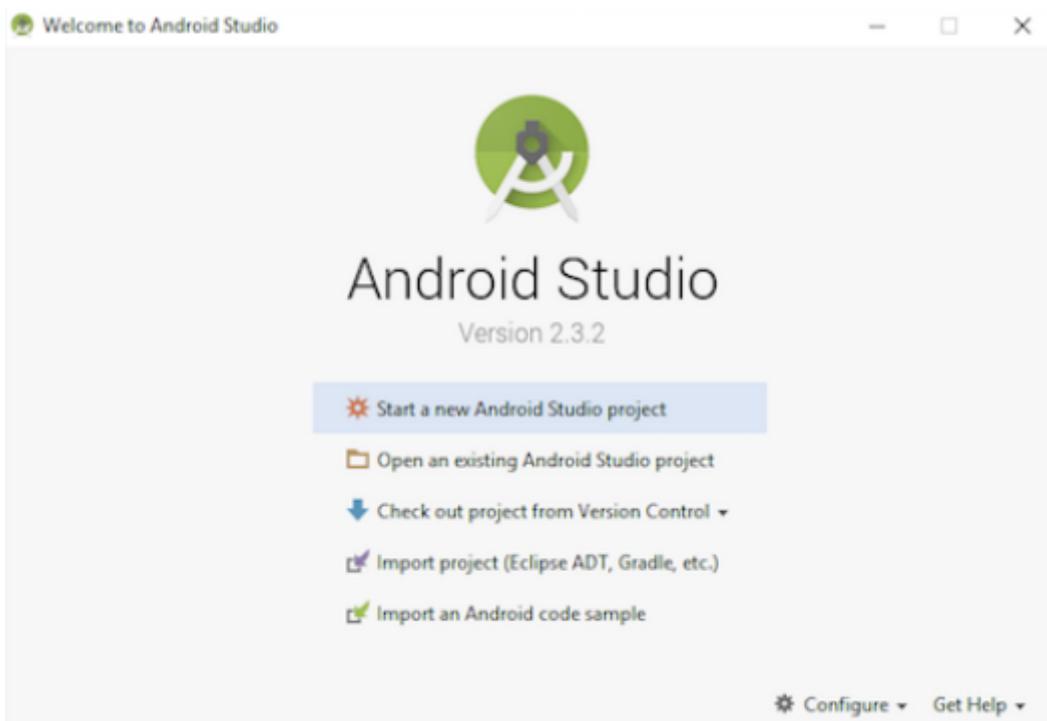
Comece instalando o Android Studio através do link:

```
https://developer.android.com/studio/index.html
```

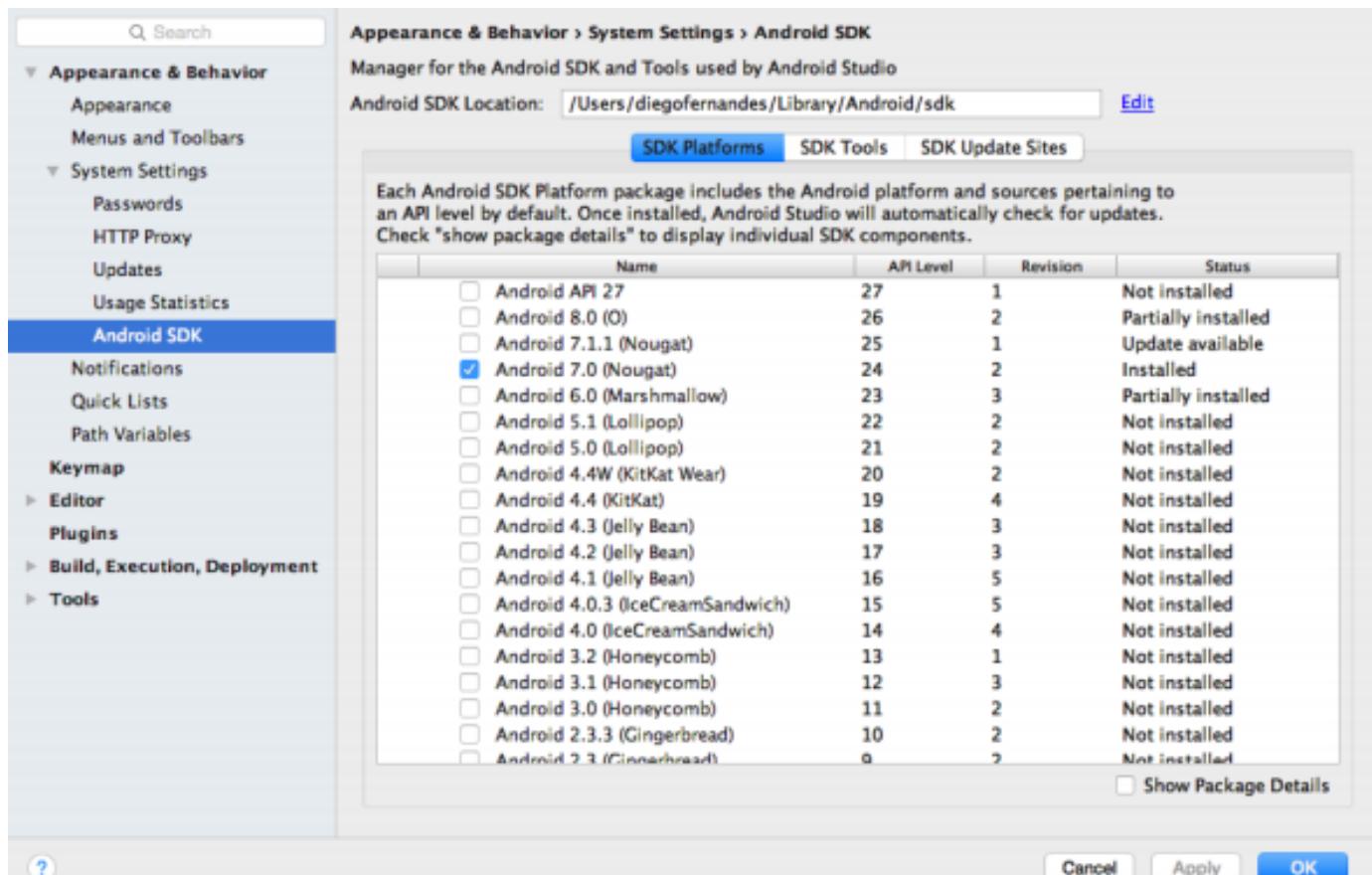
Extraia o pacote baixado em algum local do seu sistema. Assim que extraído, para executar o Android Studio, acesse a pasta de instalação pelo terminal e execute o seguinte comando a partir do caminho da pasta onde foi instalado:

[./bin/studio.sh](#)

Se a instalação ocorrer bem, você receberá uma tela como essa:



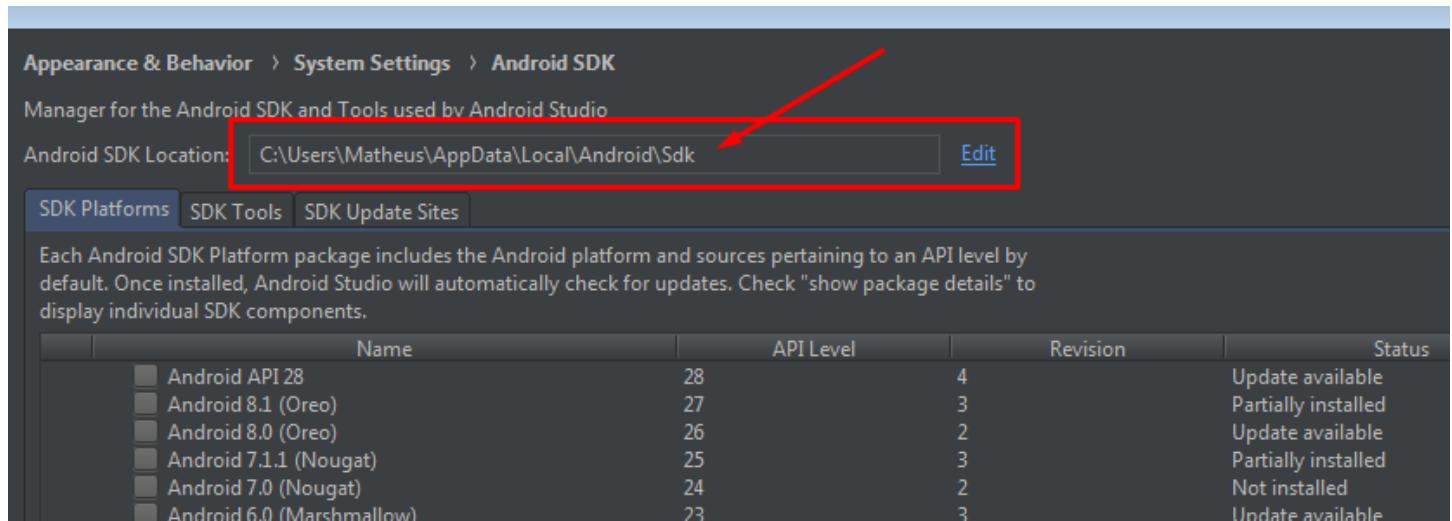
Agora, nessa tela, clique no botão “Configure” e selecione a opção “SDK Manager”. Uma tela como a seguinte deve abrir:



Certifique-se que nessa tela você tenha instalado alguma API superior à 6.0 (no meu caso a 7.0 está instalada). Na aba “SDK Tools”, certifique-se que as seguintes opções estão instaladas. Caso alguma delas ainda não esteja instalada, selecione-a e clique no botão “Apply” no final da tela para realizar o download.

	Name	Version	Status
-	Android SDK Build-Tools 27		Update Available: 27.0.0
<input type="checkbox"/>	CPU Debugging tools		Not Installed
<input type="checkbox"/>	CMake		Not Installed
<input checked="" type="checkbox"/>	LLDB		Not Installed
<input type="checkbox"/>	Android Auto API Simulators	1	Not installed
<input type="checkbox"/>	Android Auto Desktop Head Unit emulator	1.1	Not installed
<input checked="" type="checkbox"/>	Android Emulator	26.1.4	Installed
<input type="checkbox"/>	Android SDK Platform-Tools	26.0.0	Update Available: 26.0.2
<input type="checkbox"/>	Android SDK Tools	26.0.2	Update Available: 26.1.1
<input type="checkbox"/>	Documentation for Android SDK	1	Not installed
<input type="checkbox"/>	Google Play APK Expansion library	1	Not installed
<input type="checkbox"/>	Google Play Licensing Library	1	Not installed
<input type="checkbox"/>	Google Play services	46	Not installed
<input type="checkbox"/>	Google Web Driver	2	Not installed
<input type="checkbox"/>	Instant Apps Development SDK	1.1.0	Not installed
<input type="checkbox"/>	Intel x86 Emulator Accelerator (HAXM installer)	6.2.1	Not installed
<input type="checkbox"/>	NDK	15.2.4203891	Not installed
▼	<input checked="" type="checkbox"/> Support Repository		
	<input checked="" type="checkbox"/> ConstraintLayout for Android		Installed
	<input checked="" type="checkbox"/> Solver for ConstraintLayout		Installed
	<input checked="" type="checkbox"/> Android Support Repository	47.0.0	Installed
	<input checked="" type="checkbox"/> Google Repository	58	Installed

Agora que temos nossa SDK instalada com sucesso, precisamos configurar a variável **ANDROID_HOME** em nosso ambiente. O caminho dela é o caminho do Android sdk



Em seu terminal, abra o arquivo “`~/.bash_profile`” no seu editor de preferência e adicione as seguintes linhas no final do arquivo:

```
export ANDROID_HOME=$HOME/Android/Sdk  
export PATH=$PATH:$ANDROID_HOME/tools  
export PATH=$PATH:$ANDROID_HOME/platform-tools
```

*** Cole o valor copiado do Android Studio logo após o “`ANDROID_HOME=`“

Agora com tudo instalado e configurado precisamos instalar o emulador então pule para última etapa nesse **PDF** para o tópico **Genymotion**.

OS X - Mac

iOS

Para configurar o ambiente de iOS no OS X basta ter instalado o XCode no sistema. Caso você ainda não tenha instalado, você pode baixar o mesmo pelo link

<https://developer.apple.com/xcode/>

Com o XCode instalado, basta executar o seguinte comando na pasta de um projeto React Native para rodar o React Native no simulador de iOS:

[react-native run-ios](#)

Instalando Genymotion

Utilizaremos o Genymotion como emulador do Android.

Escolhi o Genymotion pelo fato de ser mais estável e rápido, mas você pode escolher utilizar o emulador do Android Studio como fazemos e configuramos na aula.

Antes de instalar o Genymotion vamos instalar o VirtualBox. Acesse o link:

<https://www.virtualbox.org/wiki/Downloads>

Para ambientes Linux basta executar o seguinte comando no terminal:

`sudo apt-get install virtualbox`

Agora vamos baixar o Genymotion:

<https://www.genymotion.com/fun-zone/>

Clicar no botão “**Download Genymotion Personal Edition**”.

O Genymotion vai pedir que você crie uma conta na plataforma, realize esse processo e depois você poderá baixar o software.

Em ambientes Windows, basta instalar o software e abri-lo através de suas aplicações.

Agora faça esse passo abaixo apenas em **LINUX** até a divisória:

Para **Linux** precisamos realizar um processo a mais: extraia o arquivo .bin baixado do site do Genymotion em uma pasta de sua escolha e acesse-a via terminal. Na pasta do arquivo extraído execute o seguinte comando:

```
chmod +x genymotion-2.2.2_x64.bin  
./genymotion-2.2.2_x64.bin
```

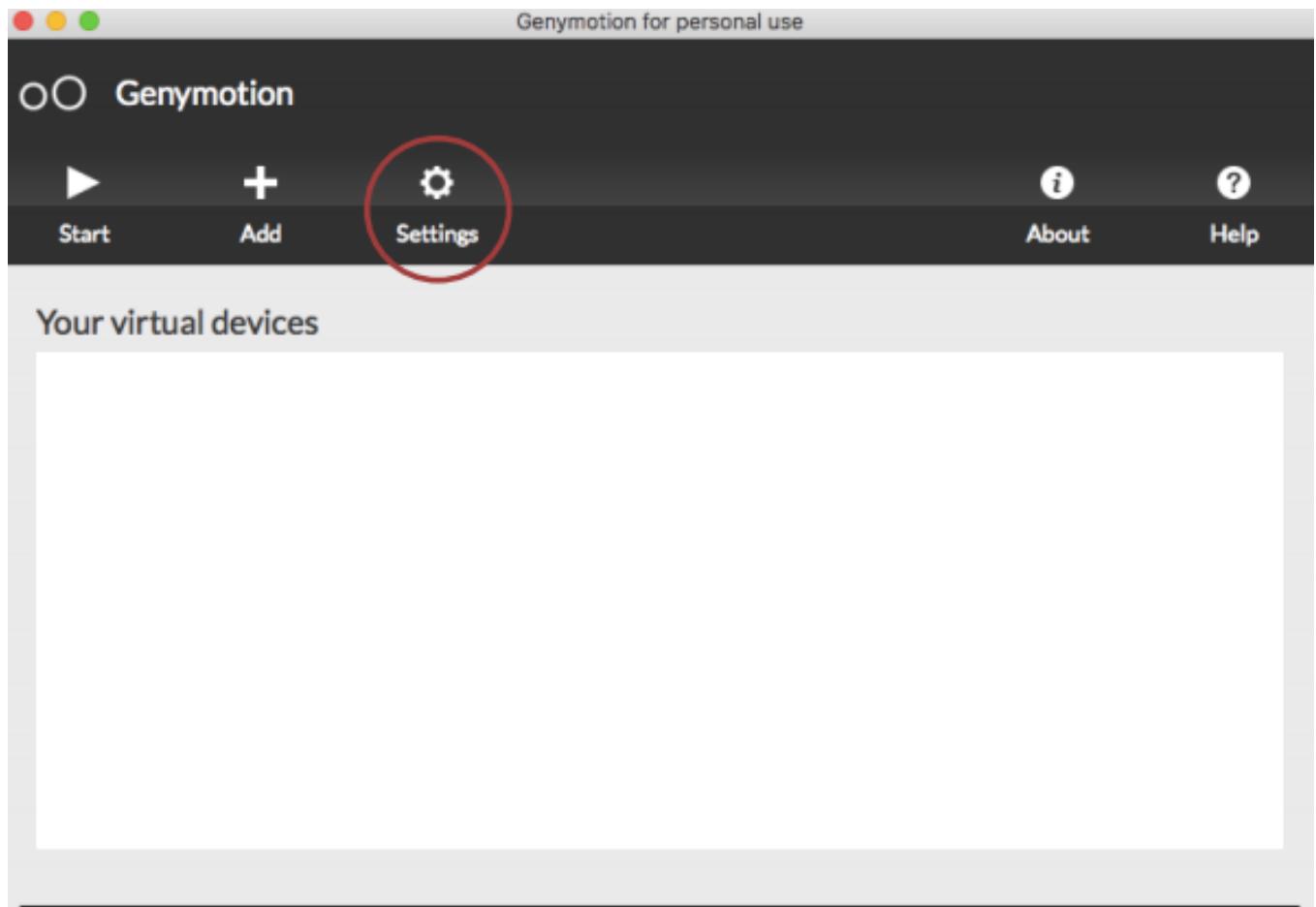
Cuidado* Altere o nome do arquivo com a versão que você baixou .

Com isso o Genymotion será instalado em seu sistema e você já pode acessá-lo a partir da pasta de instalação. Por padrão a pasta que o Genymotion é instalado é “/home/[usuario]/genymotion/“. Acesse essa pasta pelo terminal e execute o seguinte comando para abrir o Genymotion:

```
./genymotion
```

=====

Se tudo deu certo, você vai ver essa tela:



Clique no botão “Settings” e na aba “Account” faça login com sua conta criada no site do GenyMotion.

Após realizado login, ainda no menu “Settings”, na aba “ADB” precisamos informar o caminho da nossa **ANDROID_HOME** e pra isso podemos recuperar esse valor pelo Android Studio abrindo o SDK Manager:
Copie o Caminho do Android SDK LOCATION:

Appearance & Behavior > System Settings > Android SDK
Manager for the Android SDK and Tools used by Android Studio
Android SDK Location: C:\Users\Matheus\AppData\Local\Android\Sdk [Edit](#)

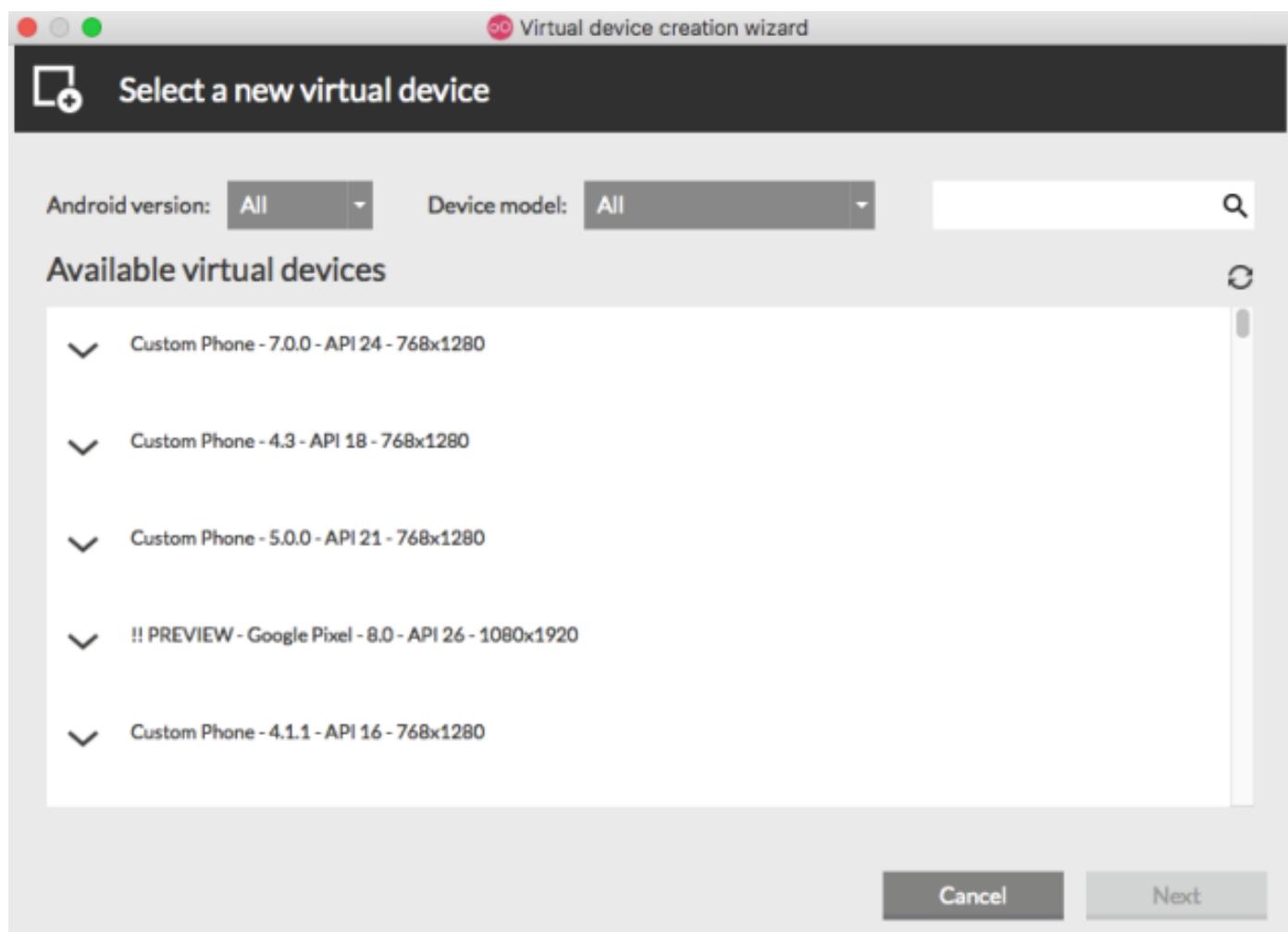
SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
<input type="checkbox"/>	Android API28	28	4	Update available
<input type="checkbox"/>	Android 8.1 (Oreo)	27	3	Partially installed
<input type="checkbox"/>	Android 8.0 (Oreo)	26	2	Update available
<input type="checkbox"/>	Android 7.1.1 (Nougat)	25	3	Partially installed
<input type="checkbox"/>	Android 7.0 (Nougat)	24	2	Not installed
<input type="checkbox"/>	Android 6.0 (Marshmallow)	23	3	Update available

Selecione a opção “Use Custom Android SDK Tools” no Genymotion e copie a variável “Android SDK Location” do Android Studio para o campo “Android SDK”.

Agora fechando as configurações, voltando a tela principal do Genymotion podemos clicar na opção “Add” que abrirá a seguinte tela:



Vou utilizar o Custom Phone 7.0.0 API 24 mas aqui você pode selecionar a opção que mais lhe agradar, tente utilizar sempre versões mais recentes do Android.
Selecione a opção e clique em Next. Depois disso ele irá te pedir um nome para o emulador, coloque o que preferir ou deixe o padrão.

Depois disso clique em Next novamente e aguarde o download das dependências. Ao acabar o processo você terá um novo emulador na lista inicial do Genymotion:



Com dois cliques em cima do Emulador agora será possível executar o ambiente Android. Caso você receba algum erro durante esse processo recomendo executar a máquina virtual do Android diretamente pelo VirtualBox pois dessa forma você terá o log de inicialização do ambiente que conterá qualquer possível erro.

Com o emulador aberto você pode realizar o run do React Native para Android através da pasta do seu projeto. Se o projeto não abrir no emulador após você ver a mensagem BUILD SUCCEEDED, procure abrir a aplicação pela tela de aplicações no android.

Agora sim já pode rodar o comando, navegue a pasta do seu app e rode: **react-native run-android**