



# Configurando ambiente de desenvolvimento React Native

Um passo a passo para instalar suas dependências e bibliotecas

Nesse documento estão descritos os passos necessários para realizar a instalação das dependências e bibliotecas assim como o emulador para configuração do ambiente de desenvolvimento em React Native. Durante todo o processo não utilizarei a ferramenta “Expo” para instalação, porém a mesma se encontra disponível caso o desenvolvedor opte por utilizá-la.

## Windows

### iOS

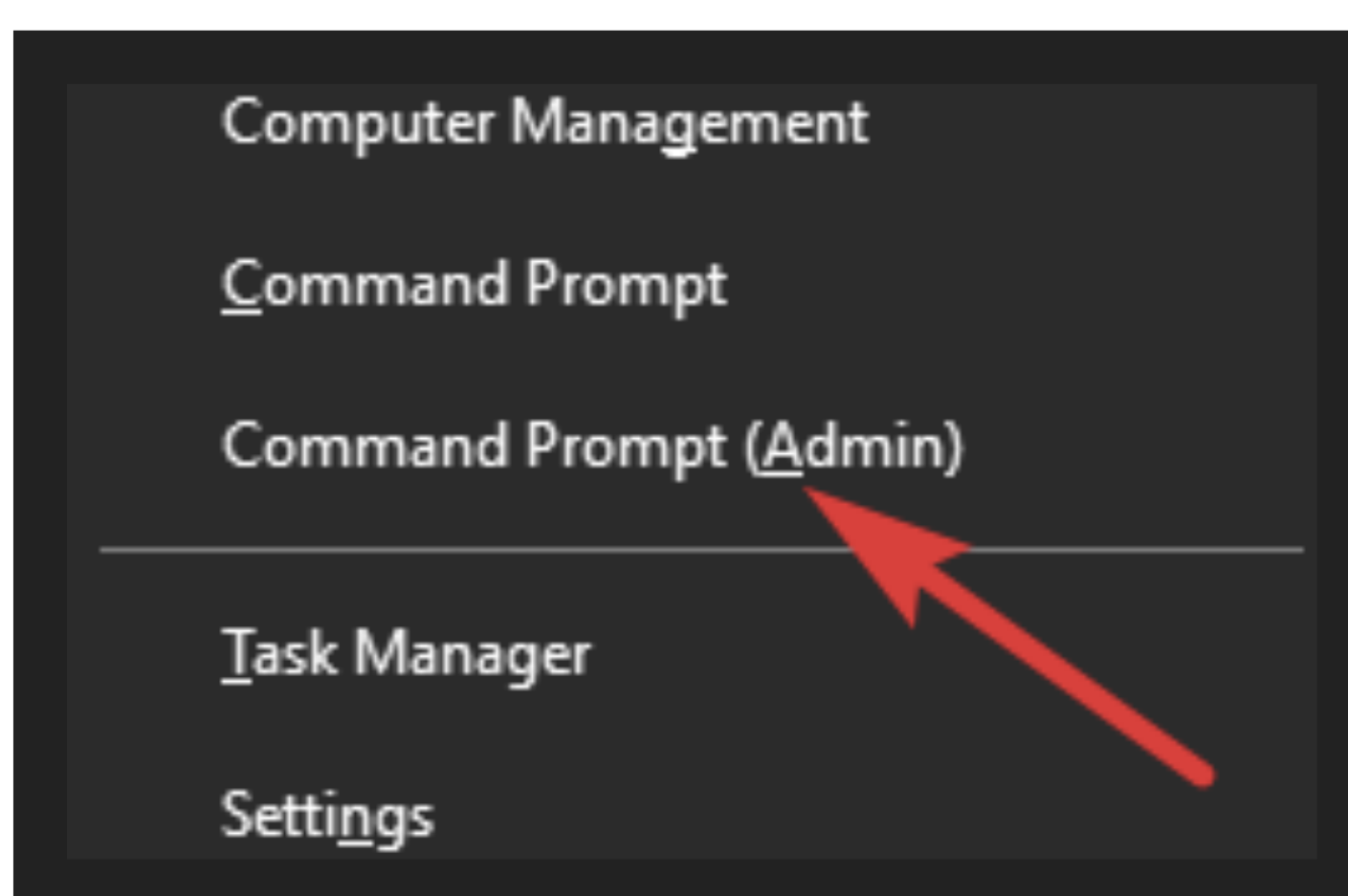
Infelizmente ainda não é possível configurar ambiente de desenvolvimento iOS no Windows, por isso, passe para o ambiente Android.

### Android

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

#### Instalando Chocolatey

Para instalar as libs no Windows, vamos utilizar um Package Manager chamado Chocolatey. Execute o prompt de comando como administrador utilizando a tecla “Windows + X” ou clicando com o botão direito sobre o botão “Iniciar”:



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):

#### cmd.exe

<https://gist.githubusercontent.com/diego3g/7596b2d4faa49adb3e4d3c24ef0504d2/raw/68379a5cff446da972f4ce501d93fbd9ed04feb0/gistfile1.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, com a devida permissão, execute o próximo comando:

<https://gist.githubusercontent.com/diego3g/40a67dd23a0b412f2fd5771ff6fd9cc6/raw/3086e8082ee6c88471b7c27fc3baf9073cb72766/chocolatey2.sh>

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

```
choco
```

Com o Chocolatey instalado, vamos instalar o **Node**, **Python2** e a **JDK8** (Java Development Kit 8):

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

obs: se você já tiver instalado o NodeJS no seu Windows, certifique-se que sua versão é superior à 4. Se tiver o JDK instalado em seu sistema, certifique-se que sua versão é superior à 8.



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

obs: se você obter um erro “Cannot find module ‘npmlog’”, execute o comando “curl -O -L https://npmjs.org/install.sh | sudo sh”.

Com o NodeJS instalado, vamos instalar o CLI (Command Line Interface) do React Native:

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

```
react-native -h
```

Android Studio

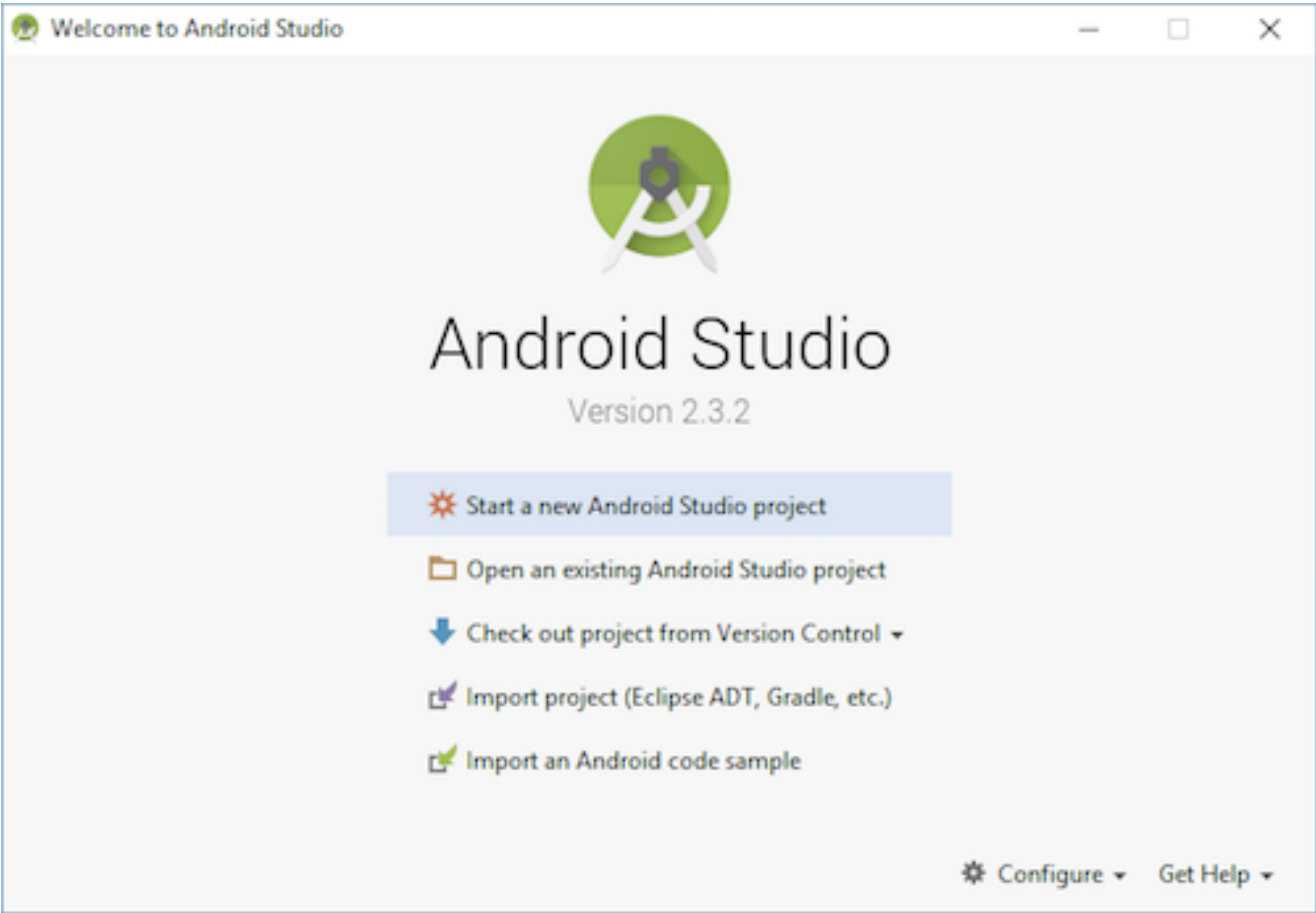
Para emular nossa aplicação vamos precisar de um sistema Android, você pode optar por utilizar seu próprio smartphone ou configurar um emulador no Windows, caso opte por utilizar o próprio dispositivo físico, basta conectá-lo via porta USB e pular o próximo passo.

Comece instalando o Android Studio através do link: <https://developer.android.com/studio/index.html>

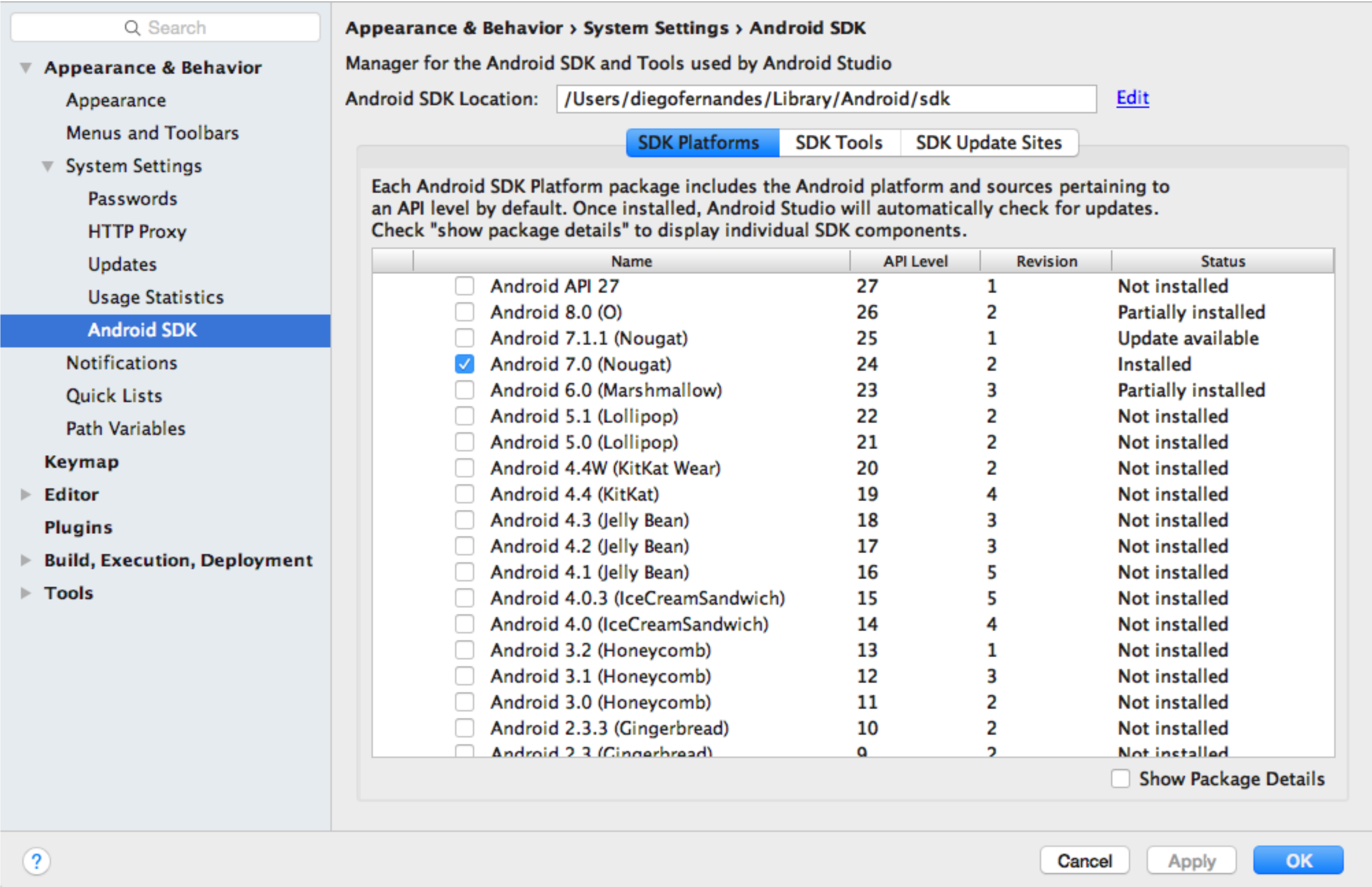
Assim que feito o Download, execute o instalador. No passo “Tipo de instalação” selecione “Custom” ou “Personalizada” e selecione os seguintes componentes:

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

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

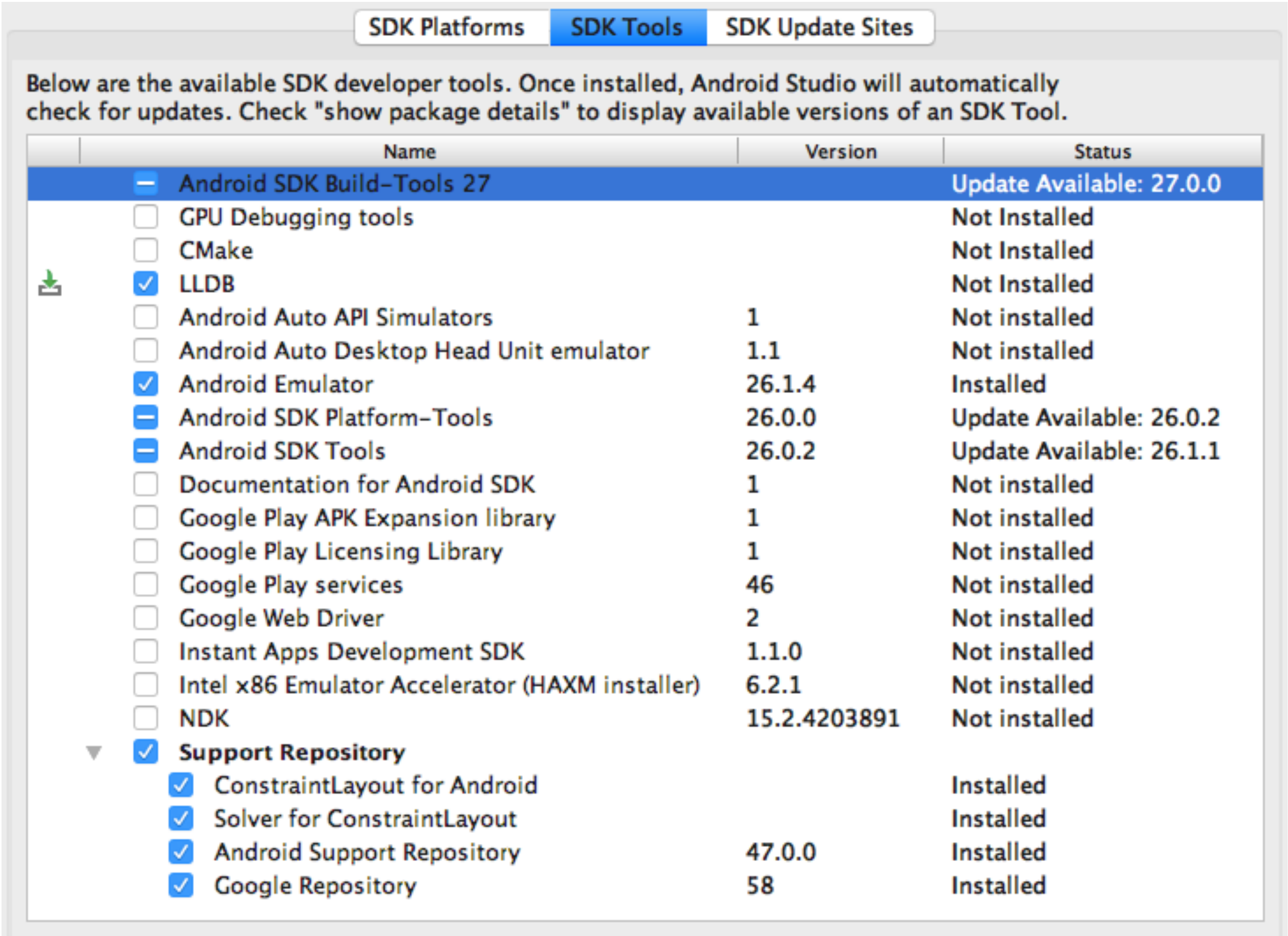


Agora, nessa tela, clique no botão “Configure” e selecione a opção “SDK Manager”. Uma tela como a seguinte deve abrir:



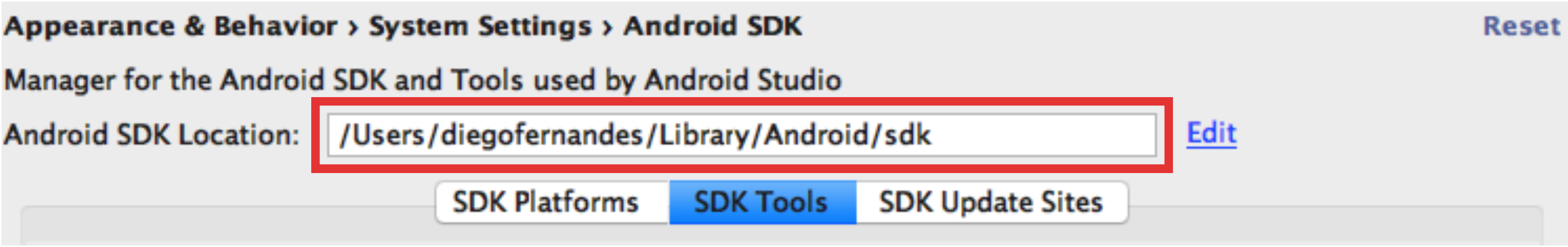


Certifique-se que nessa tela você tenha instalado alguma API superior à 6.0 (no meu caso a 7.0 está instalada). Na aba “SDK Tools”, certifique-se que as seguintes opções estão instaladas. Caso alguma delas ainda não esteja instalada, selecione-a e clique no botão “Apply” no final da tela para realizar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

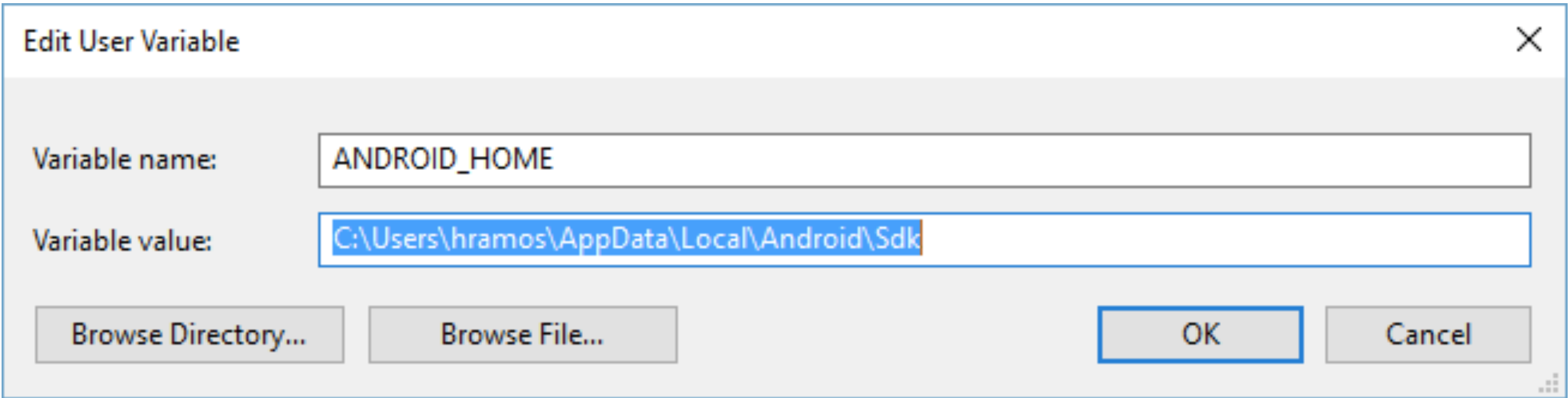


### Configurando ANDROID\_HOME

Agora que temos nossa SDK instalada com sucesso, 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 a API configurada precisamos instalar o emulador e para você pode pular para a última etapa nesse PDF chamada “Configurando Genymotion”.  
Apos configurar, com o emulador aberto, basta executar o seguinte comando:

```
react-native init NomeDoProjeto
cd NomeDoProjeto
react-native run-android
```

obs: Para rodar o comando run-android você precisa sempre estar na pasta de um projeto React Native, você pode substituir NomeDoProjeto pelo nome do seu projeto.



# Linux

## iOS

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

## Android

Para configurar o ambiente Android no Linux, vamos precisar instalar 3 dependências: **Node, JDK e Android Studio**

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 (Command Line Interface) 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
```

Instale o JDK8 utilizando o comando:

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

### Android Studio

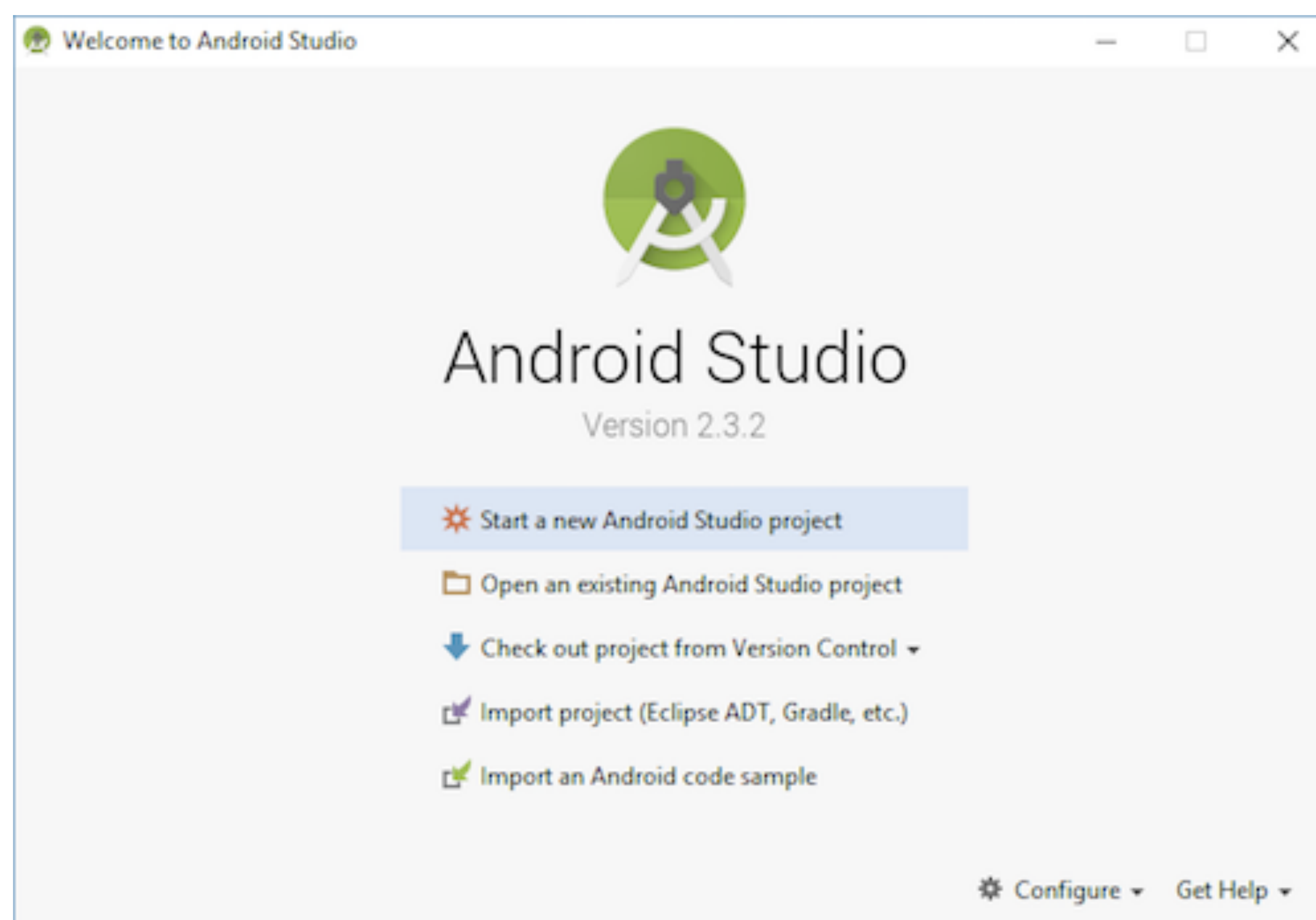
Para emular nossa aplicação vamos precisar de um sistema Android, você pode optar por utilizar seu próprio smartphone ou configurar um emulador no Linux, caso opte por utilizar o próprio dispositivo físico, basta conectá-lo via porta USB e pular o próximo passo.

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:

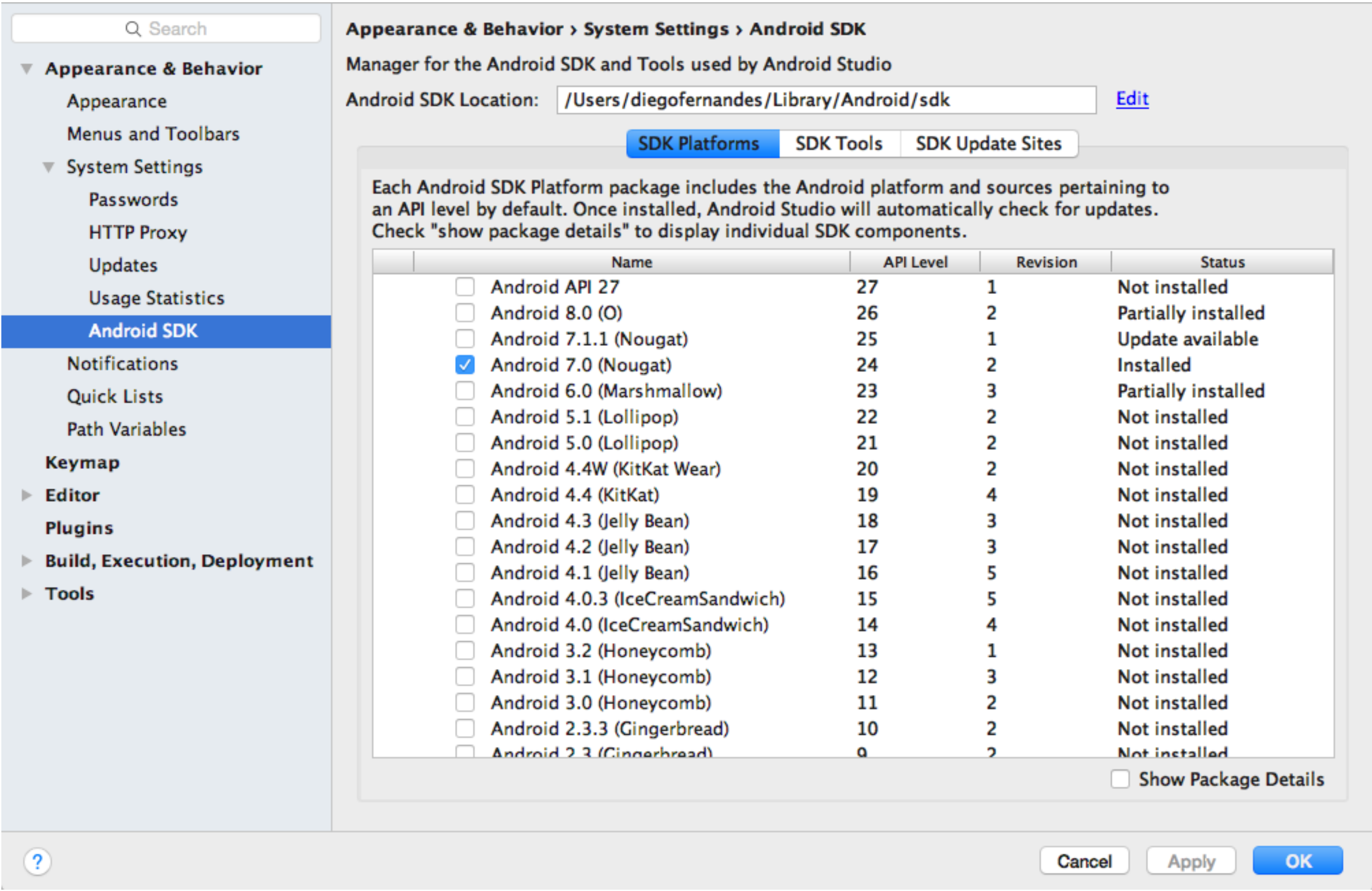
```
./bin/studio.sh
```

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

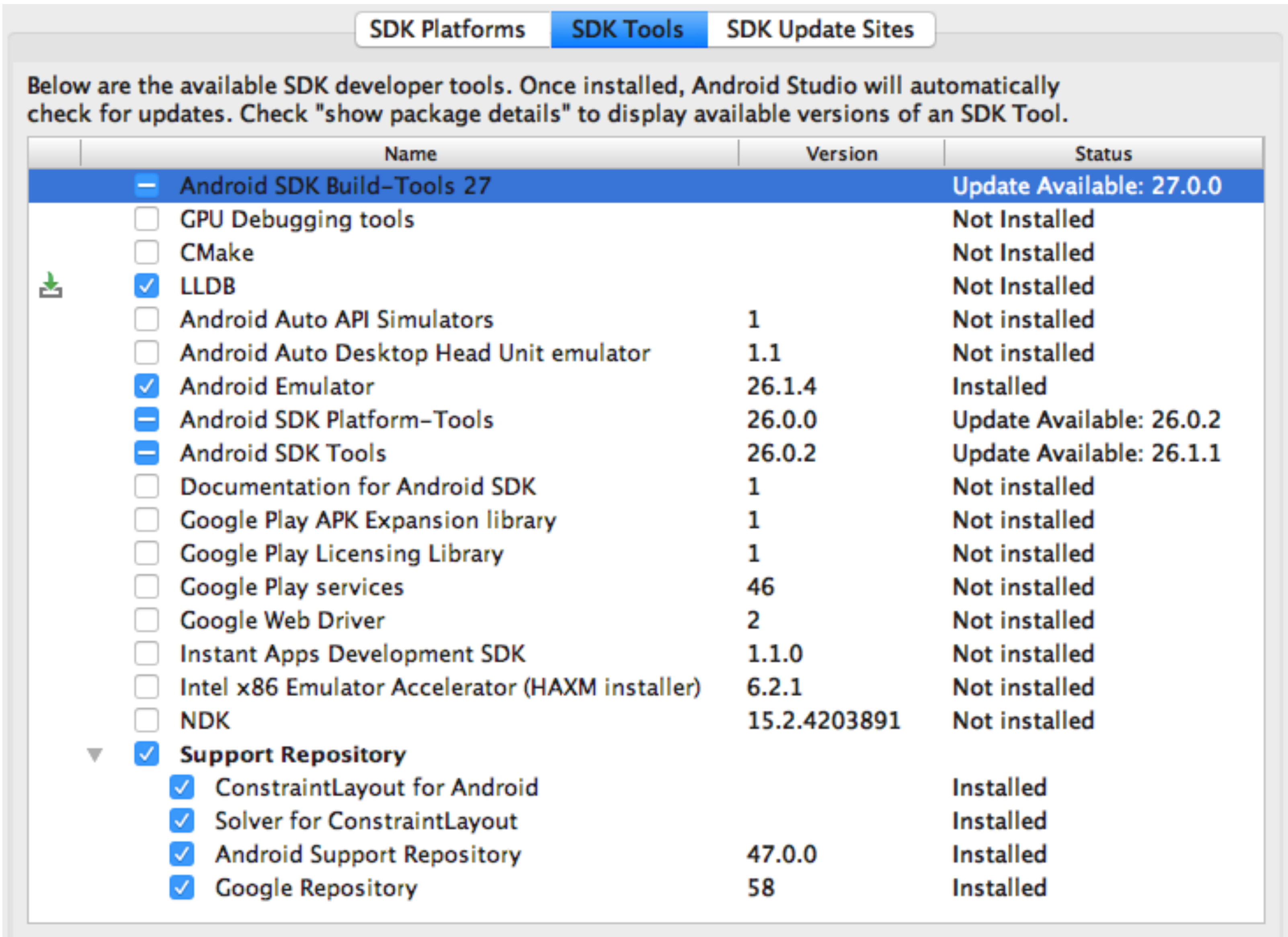


Agora, nessa tela, clique no botão “Configure” e selecione a opção “SDK Manager”. Uma tela como a seguinte deve abrir:



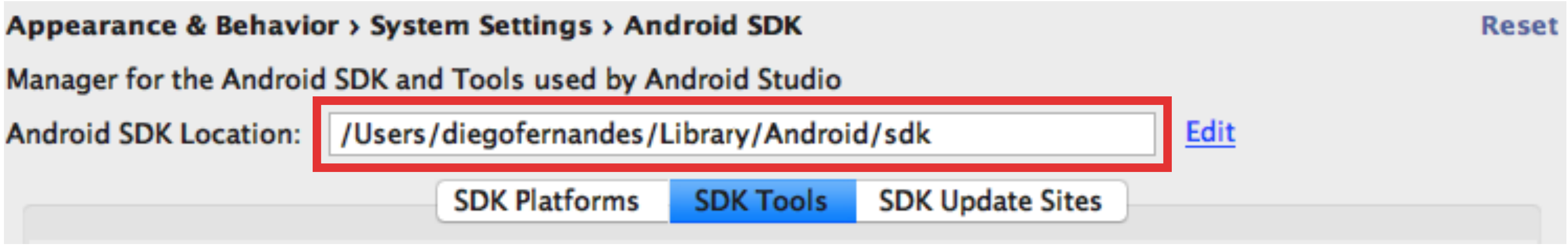


Certifique-se que nessa tela você tenha instalado alguma API superior à 6.0 (no meu caso a 7.0 está instalada). Na aba “SDK Tools”, certifique-se que as seguintes opções estão instaladas. Caso alguma delas ainda não esteja instalada, selecione-a e clique no botão “Apply” no final da tela para realizar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



Agora que temos nossa SDK instalada com sucesso, 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:

### Configurando ANDROID\_HOME



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
```

**ATENÇÃO:** Cole o valor copiado do Android Studio logo após o “ANDROID\_HOME=”.

obs: se você estiver utilizando outra instalação de terminal como o “Oh My Zsh”, você precisará adicionar essas variáveis ao arquivo de configuração desse terminal e não ao .bash\_profile.

Agora com a API configurada precisamos instalar o emulador e para você pode pular para a última etapa nesse PDF chamada “Configurando Genymotion”.

Com o emulador aberto, basta executar o seguinte comando:

```
react-native init NomeDoProjeto
cd NomeDoProjeto
react-native run-android
```

obs: Para rodar o comando run-android você precisa sempre estar na pasta de um projeto React Native, você pode substituir NomeDoProjeto pelo nome do seu projeto.



# 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
```

## Android

Para configurar o ambiente Android no Linux, vamos precisar instalar 4 dependências: **Node**, **Watchman**, **JDK** e **Android Studio**. Para boa parte desses pacotes utilizaremos o Homebrew para instalação.

### Instalando Homebrew

O Homebrew é um gerenciador de pacotes para OS X muito famoso e útil. Vamos instalá-lo em nosso sistema como seguinte comando:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

obs: o Ruby já vem instalado com o sistema OS X, se em algum caso não tiver instalado no seu sistema, procure seguir o tutorial de instalação através do site do Ruby

Com o Homebrew instalado, vamos instalar o NodeJS e o Watchman:

```
brew install node
brew install watchman
```

obs: se você já tiver instalado o NodeJS no seu Windows, certifique-se que sua versão é superior à 4.

Com o NodeJS instalado podemos seguir para a instalação do CLI (Command Line Interface) do React Native:

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

obs: se você obter um erro “Cannot find module ‘npmlog’”, execute o comando “curl -O -L https://npmjs.org/install.sh | sudo sh”.

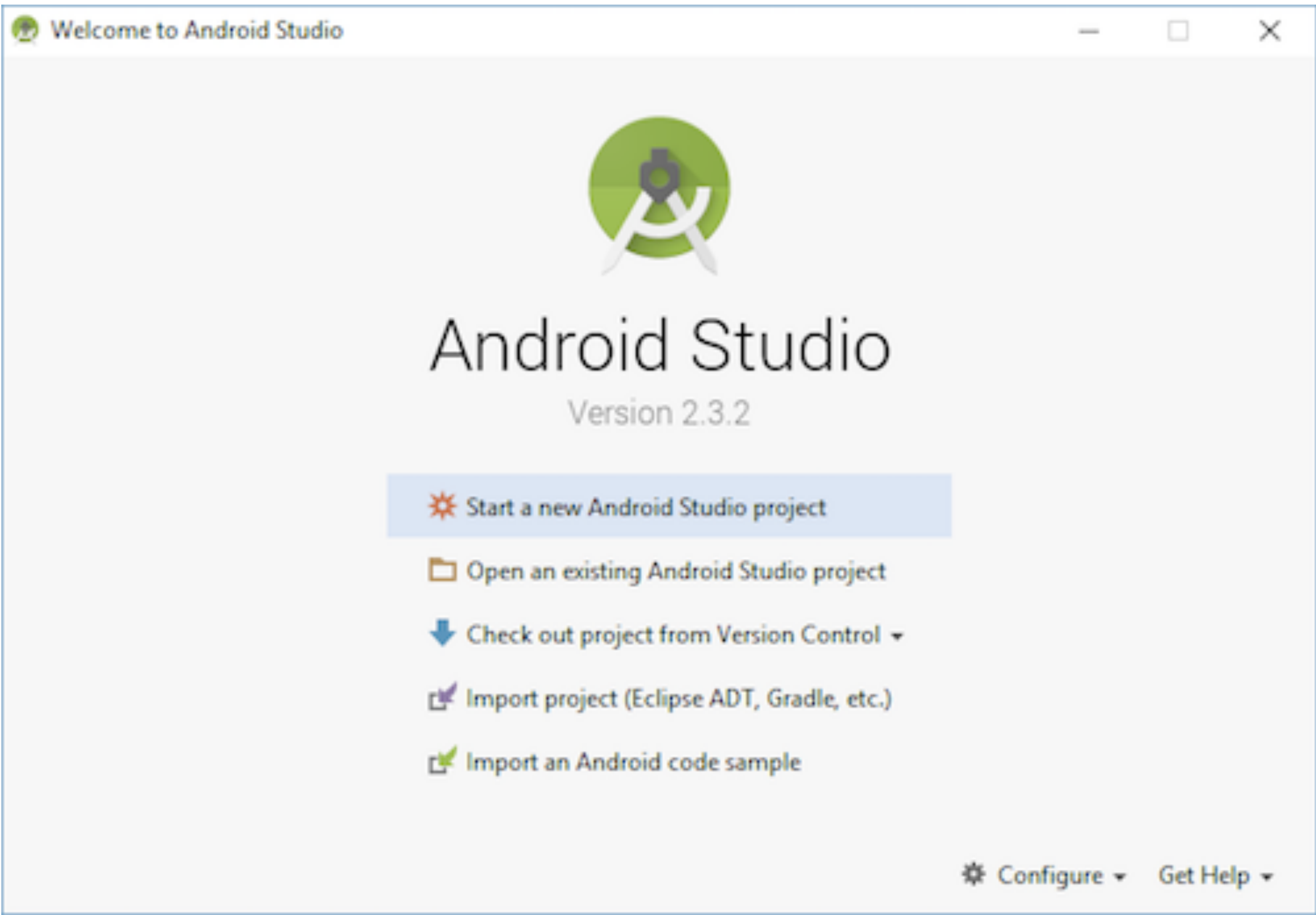
O último item a ser instalado é o JDK (Java Development Kit) do Java que pode ser baixado pelo link (Após instalado, execute o .dmg e instale seguindo os passos do instalador): <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

### 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 Mac, caso opte por utilizar o próprio dispositivo físico, basta conectá-lo via porta USB e pular o próximo passo.

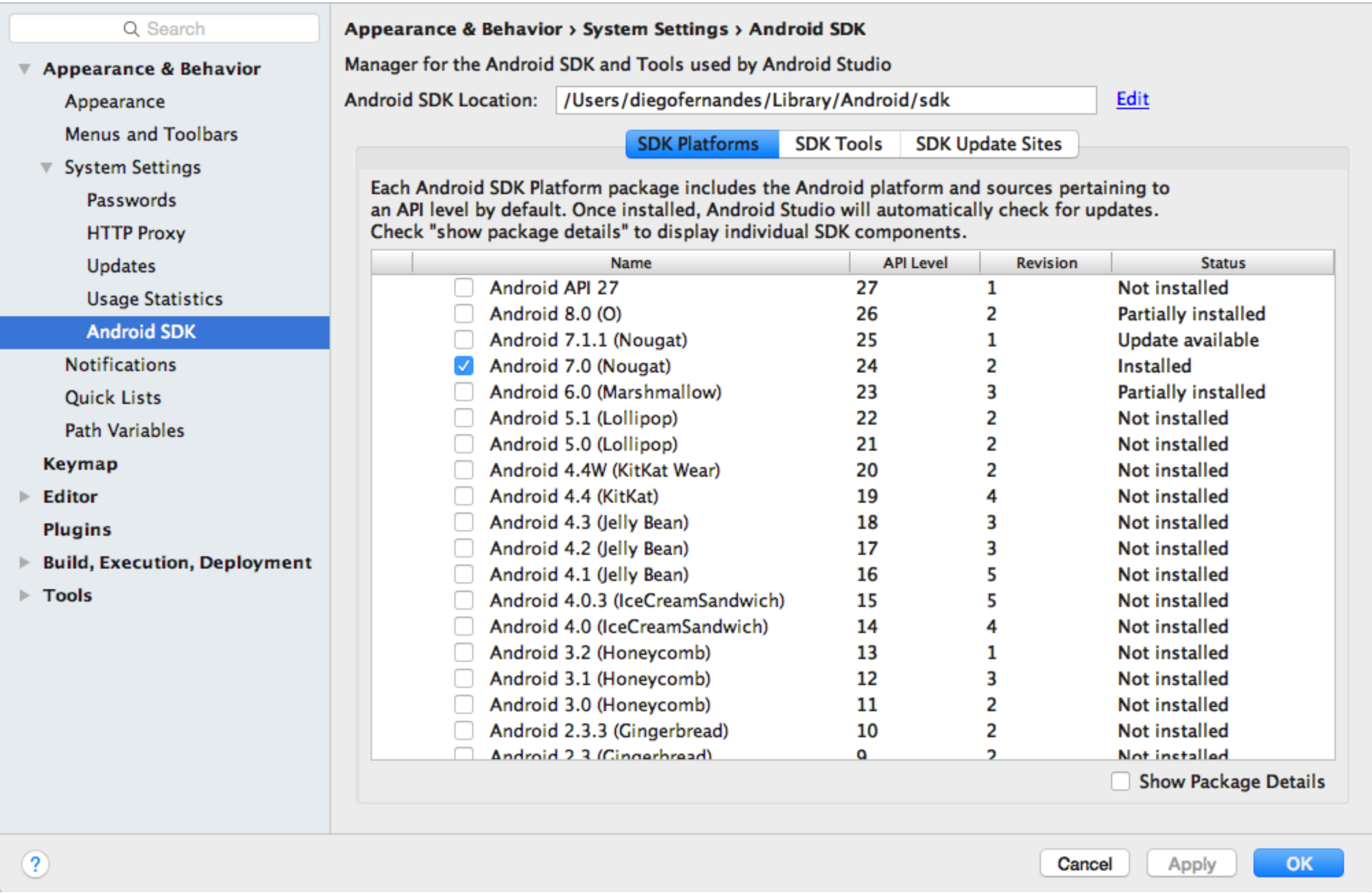
Comece instalando o Android Studio através do link: <https://developer.android.com/studio/index.html>

Assim que baixado, execute o Android Studio e uma tela como a seguir deve aparecer:

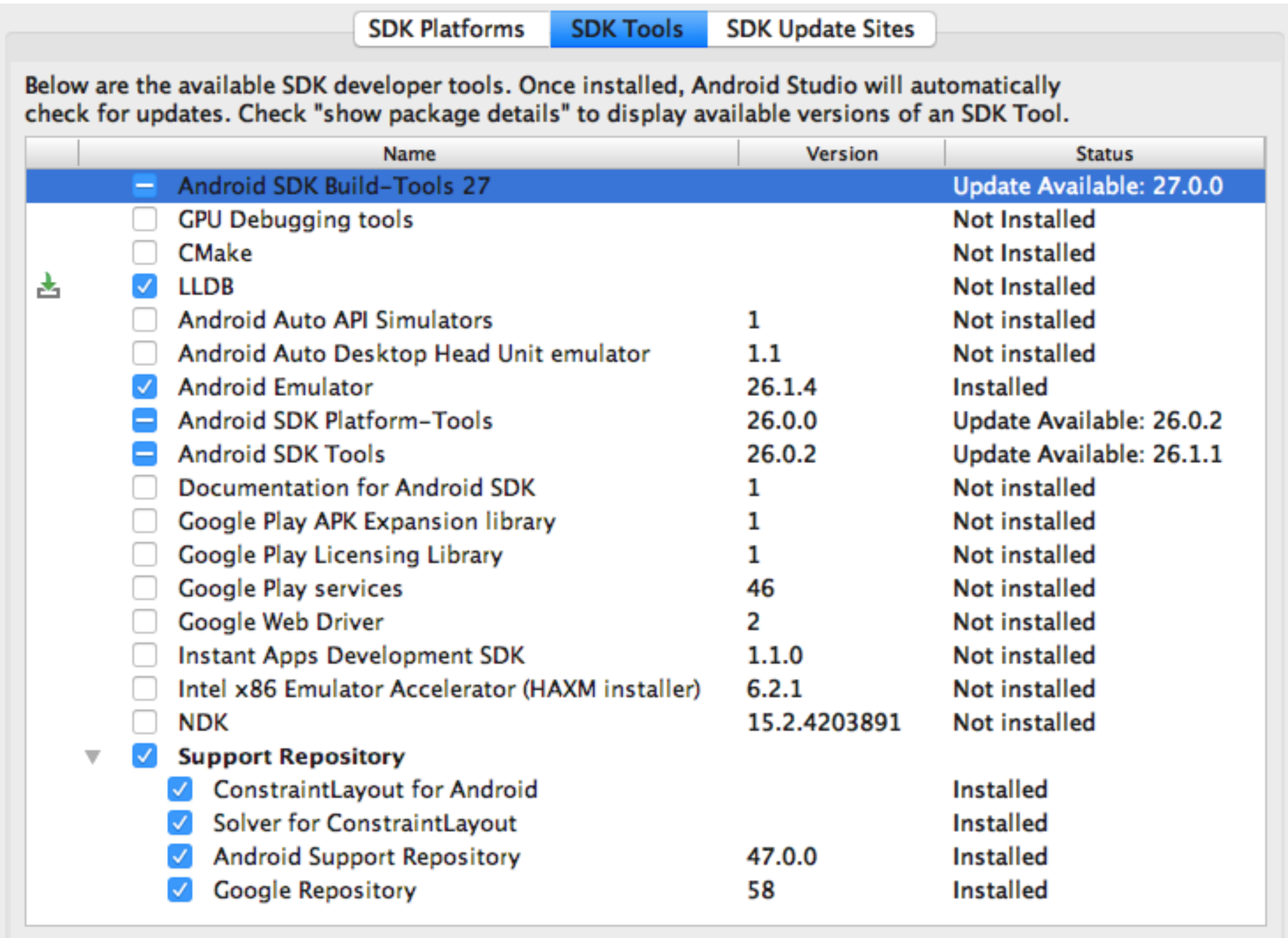


Agora, nessa tela, clique no botão “Configure” e selecione a opção “SDK Manager”. Uma tela como a seguinte deve abrir:



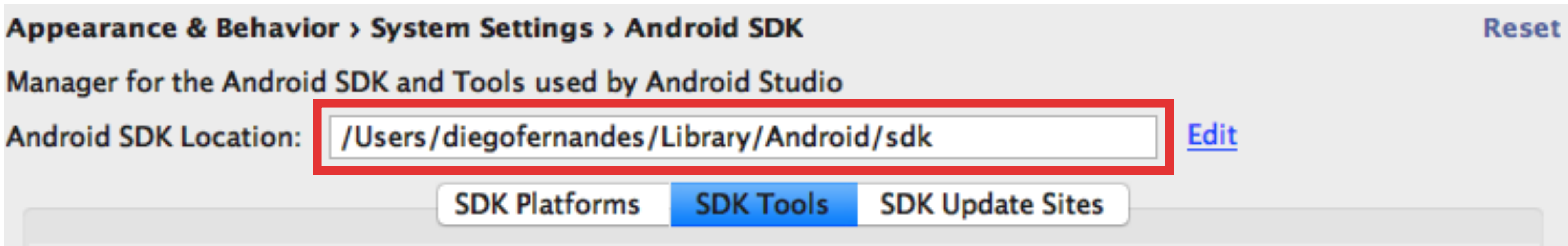


Certifique-se que nessa tela você tenha instalado alguma API superior à 6.0 (no meu caso a 7.0 está instalada). Na aba “SDK Tools”, certifique-se que as seguintes opções estão instaladas. Caso alguma delas ainda não esteja instalada, selecione-a e clique no botão “Apply” no final da tela para realizar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



Agora que temos nossa SDK instalada com sucesso, 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:

Configurando ANDROID\_HOME



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

**ATENÇÃO:** Cole o valor copiado do Android Studio logo após o “ANDROID\_HOME=“.

obs: se você estiver utilizando outra instalação de terminal como o “Oh My Zsh”, você precisará adicionar essas variáveis ao arquivo de configuração desse terminal e não ao .bash\_profile.

Agora com a API configurada precisamos instalar o emulador e para você pode pular para a última etapa nesse PDF chamada “Configurando Genymotion”.

Com o emulador aberto, basta executar o seguinte comando:

```
react-native init NomeDoProjeto
cd NomeDoProjeto
react-native run-android
```

obs: Para rodar o comando run-android você precisa sempre estar na pasta de um projeto React Native, você pode substituir NomeDoProjeto pelo nome do seu projeto.



# Configurando 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.

Antes de instalar o Genymotion vamos instalar o VirtualBox. Para ambientes **Mac OSX ou Windows** 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 acessar o site <https://www.genymotion.com/fun-zone/> e 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 **Mac OSX ou Windows**, basta instalar o software e abrí-lo através de suas aplicações.

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:

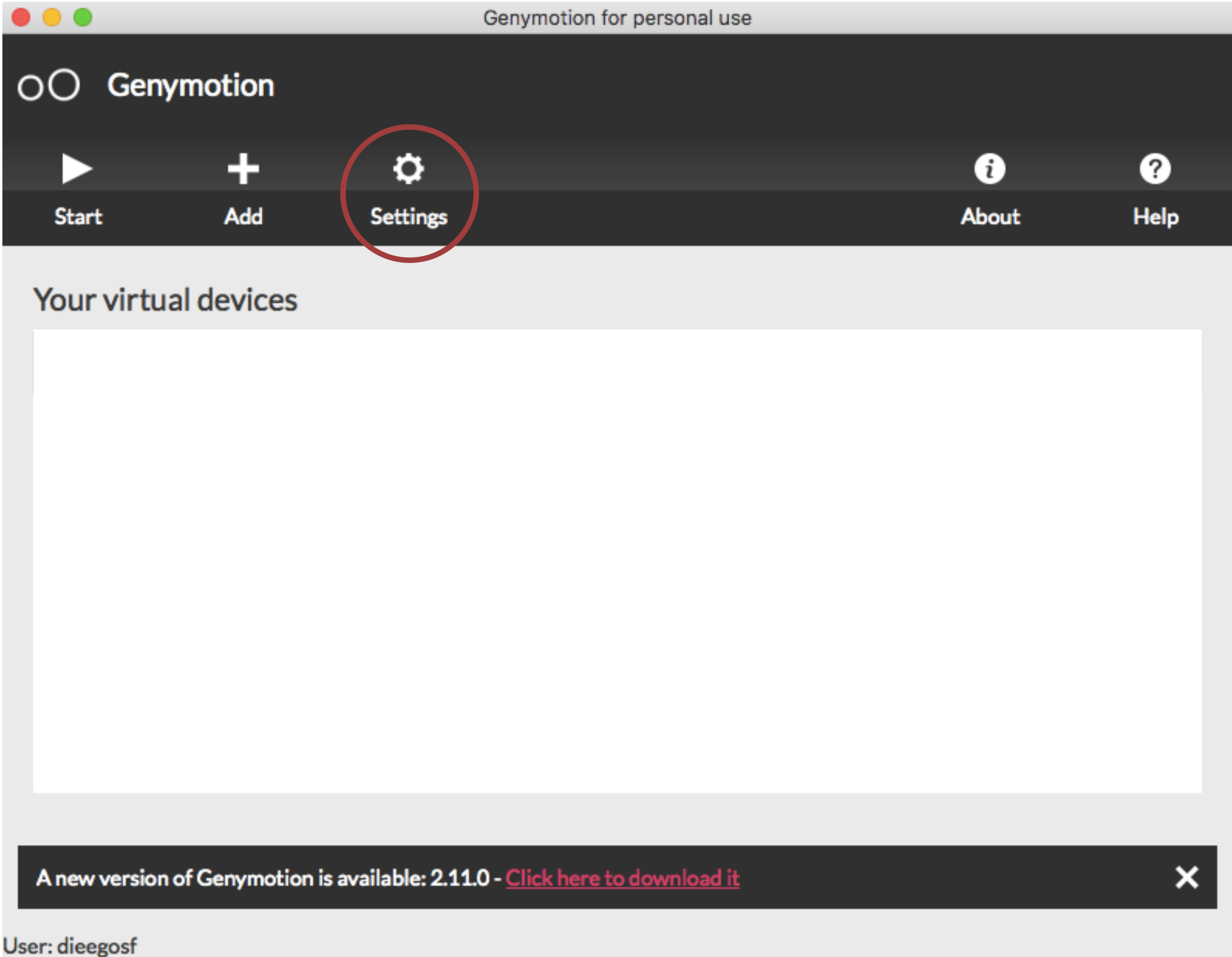
```
chmod +x genymotion-2.2.2_x64.bin
./genymotion-2.2.2_x64.bin
```

**ATENÇÃO:** Altere o nome do arquivo com a versão que você baixou (pode utilizar o TAB para completar o nome quando estiver digitando)

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:

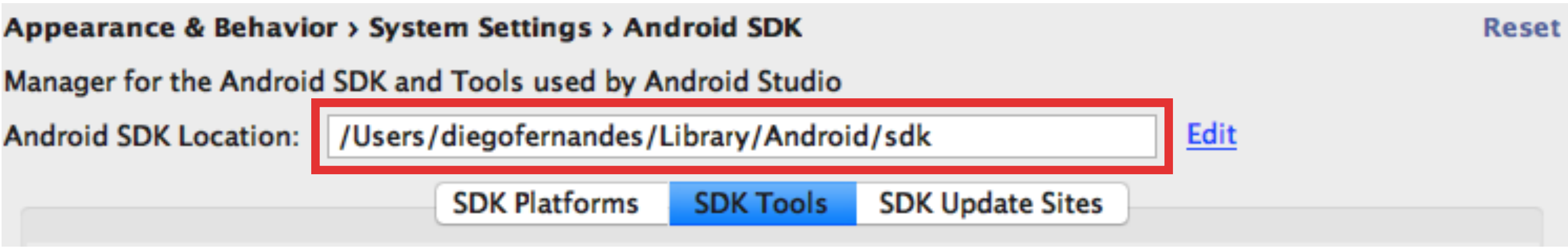
```
./genymotion
```

Se tudo ocorreu bem, você verá uma tela como a seguinte:



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

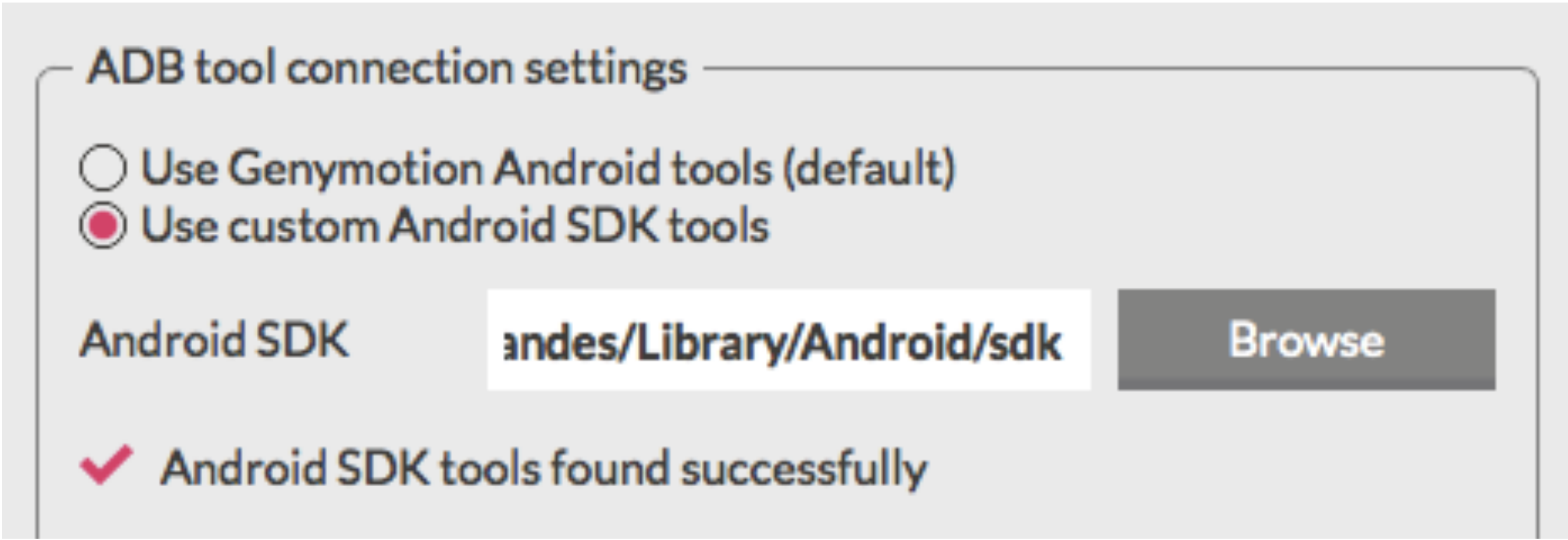
Após realizado login, ainda no menu “Settings”, na aba “ADB” precisamos informar o caminho da nossa ANDROID\_HOME e pra isso podemos recuperar esse valor pelo Android Studio abrindo o SDK Manager:



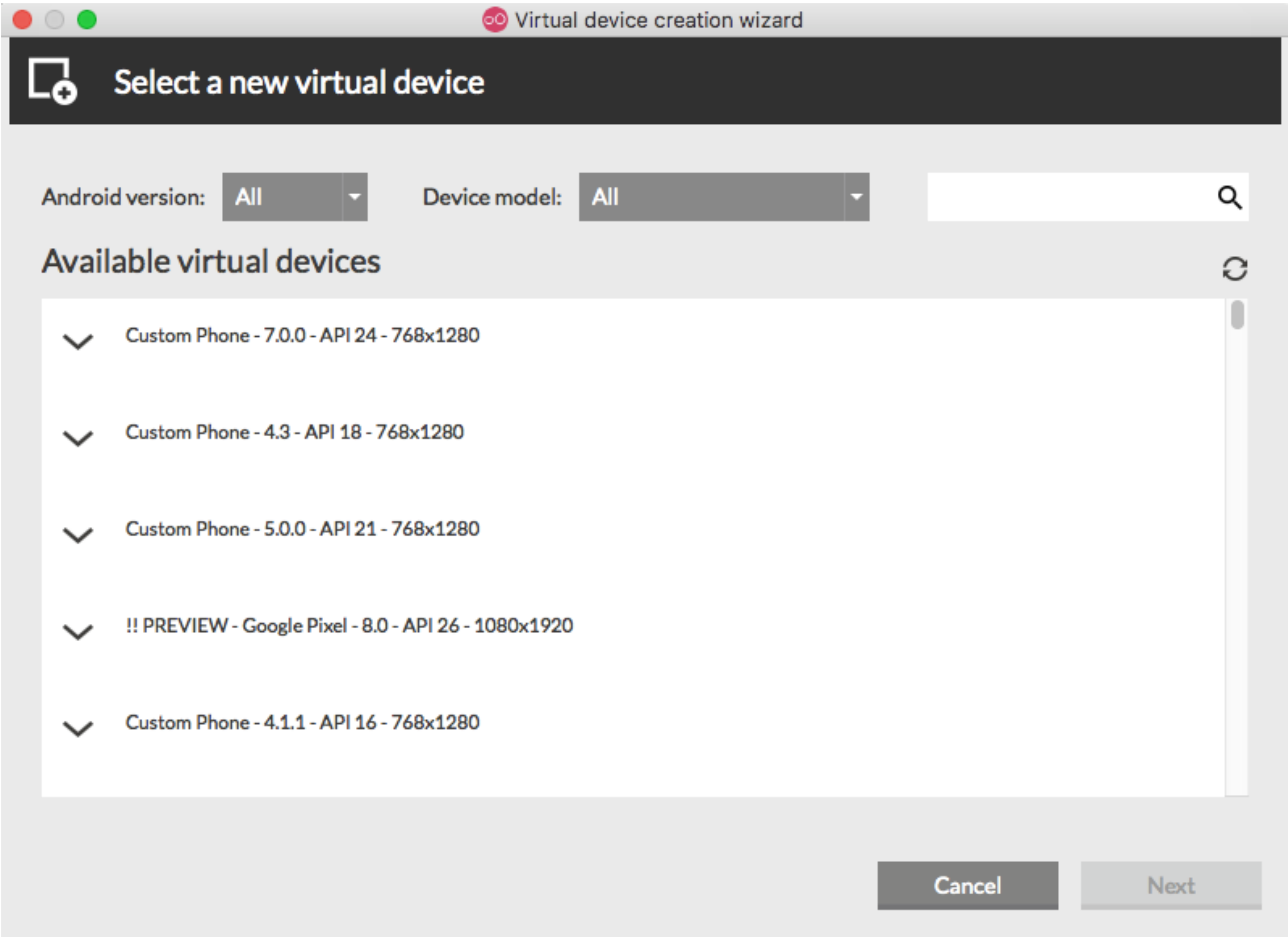
Selecione a opção “Use Custom Android SDK Tools” no Genymotion e copie a variável “Android SDK Location” do Android Studio para o campo “Android SDK”:



Sua configuração deve ficar parecida com essa:



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 conterà 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, geralmente aberta por um ícone ☰:

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