03

Preparando ambiente Windows

Preparando o ambiente

Para conseguir criar projetos, desenvolver nossos apps e testar seus comportamentos precisamos primeiramente entender quais softwares e ferramentas são necessários para configurar corretamente nosso ambiente de desenvolvimento.

Mãos à obra

A preparação do ambiente de desenvolvimento para o React Native além de levar em conta as diferenças entre os sistemas operacionais da máquina de desenvolvimento, também precisa se ater a detalhes de cada uma das plataformas do mundo mobile onde queremos testar e implantar nossas apps (Android/IOS). Por esse motivo, nossa etapa de preparação do ambiente está divida primeiramente por plataforma da máquina de desenvolvimento, seguindo para o ambiente das plataformas mobile.

Windows

IMPORTANTE: Usando sistemas Windows não é possível desenvolver apps com código nativo para a plataforma IOS.

Preparando ambiente para desenvolvimento Android com React Native

Vamos começar instalando todos os recursos necessários para rodar apps Android utilizando o React Native. Todos os passos dessa configuração estão disponíveis na documentação do React Native, que pode ser acessada através deste link (https://facebook.github.io/react-native/docs/getting-started.html).

Node js, Python 2 e Java

Para começar precisaremos contar com o runtime do Node js (https://nodejs.org/en/), Python 2 e o Java Development Kit na versão 8 ou mais recente. Assim como a própria documentação do React Native, recomendamos que você instale essas dependências com o auxílio do gerenciador de pacotes para Windows, Chocolatey (https://chocolatey.org/).

Após a instalação do Chocolatey, clique com o direito no seu prompt de comando e execute-o como administrador. A seguir, execute o seguinte comando:

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

Assim o gerenciador de pacotes já vai providenciar as instalações

Caso já tenha o node instalado em sua máquina certifique-se de que a versão instalada é a 4 ou mais recente, como recomendado na própria documentação. Recomendamos o uso da versão 6.x.x, para já podermos contar com algumas das features do ES6, por exemplo.

React Native

Atualmente precisamos fazer a instalação de nenhum CLI global para executarmos comandos React-native, isso porque a equipe do React criou uma lib executável através do `npx`, que vem instalado com o Node. Nesse caso só precisamos chamar os comandos do react native com o npx.

```
npx react-native init
npx react-native start
npx react-native run-ios
npx react-native run-android
```

Caso você queira instalar o CLI do react-native você pode faze-lo com o seguinte comando `npm install -g react-native-cli`. Nesse caso, basta não colocar `npx` em frente aos comandos que quer executar.

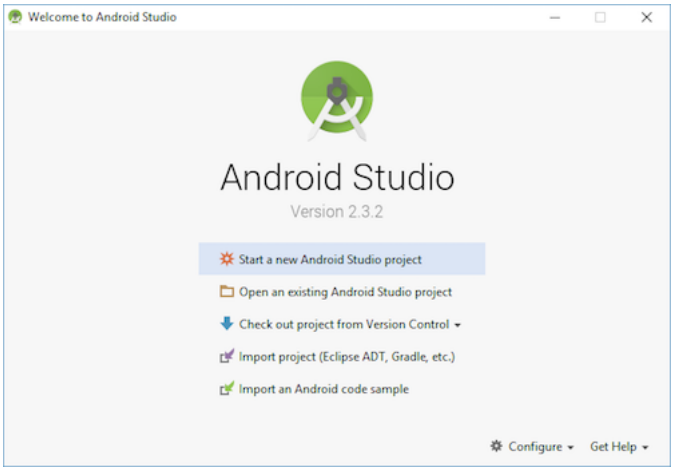
> Recomenda-se a utilização do Node Package Manager na versão 4. Você também pode (e é recomendável) utilizar a ferramenta Yarn (https://yarnpkg.com). Yarn é um gerenciador de pacotes criado também pelo Facebook que já conta com uma série de otimizações para facilitar o gerenciamento das dependências nos seus projetos que usam ferramentas da própria empresa como React, React Native, Jest, Watchman, etc. Para instalar o Yarn utilizando o Homebrew digite o seguinte comando no terminal: `brew install yarn`

Android Studio

Seguindo em frente precisaremos também do ambiente de desenvolvimento Android configurado, portanto, vamos baixar e instalar também a ferramenta Android Studio e as SDK Tools. Baixe e instale o Android Studio (https://developer.android.com/studio/index.html), selecione *"Custom"* quando perguntado sobre o tipo de instalação desejado e certifique-se de marcar as seguintes opções no instalador da ferramenta antes de clicar em *"Next"* e instalar efetivamente os componentes:

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

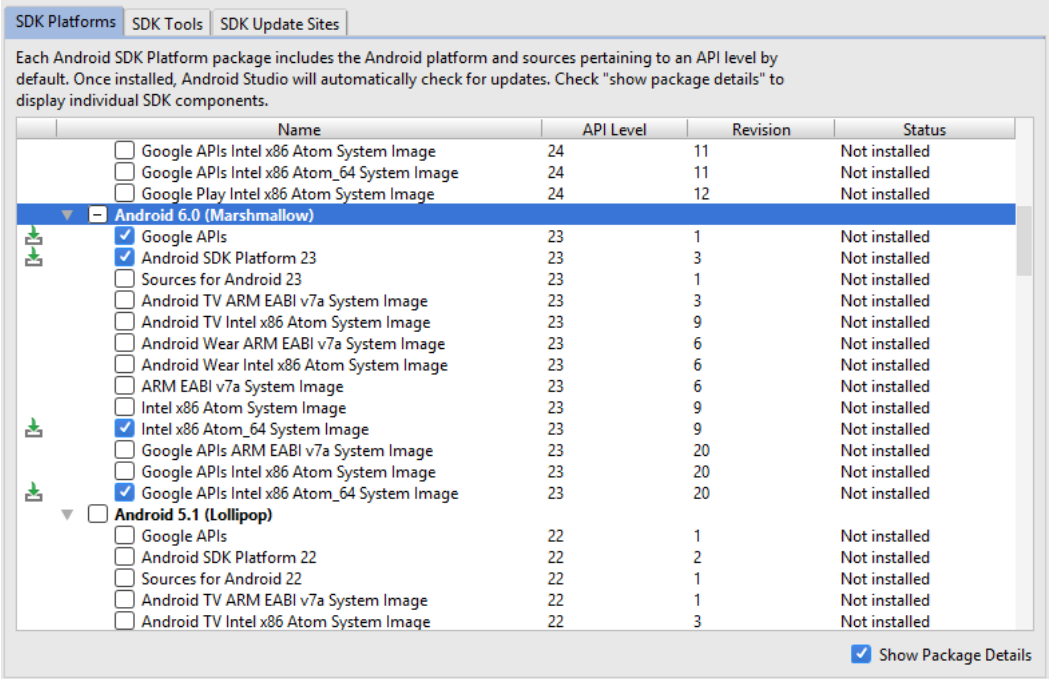
Após a instalação, será exibida uma tela de boas vindas do Android Studio.



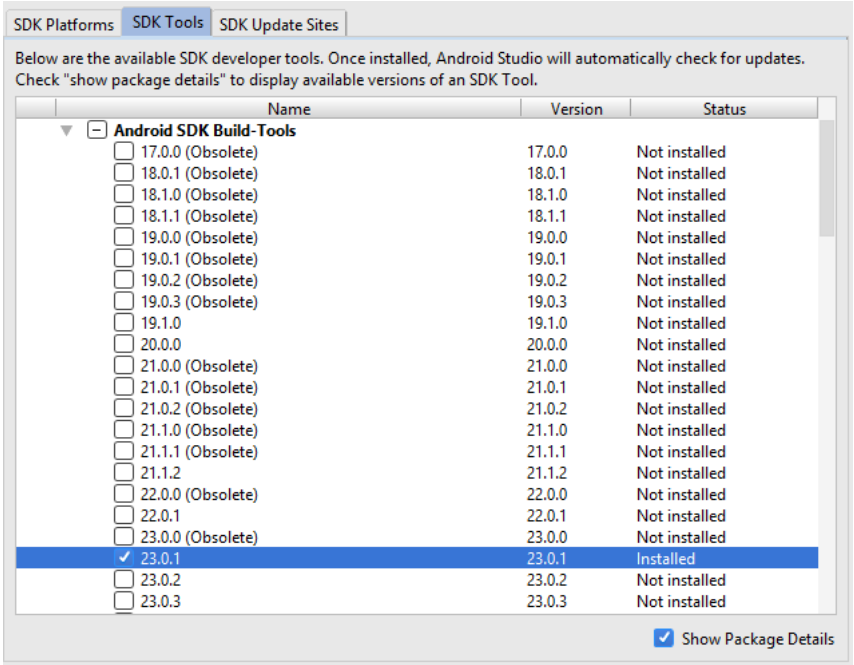
O Android Studio já instala por padrão a última versão do SDK do Android, porém para desenvolver apps nativas para o Android com React Native, precisaremos instalar adicionalmente o SDK na versão Android 6.0 (Marshmallow). Podemos adicionar SDKs adicionais no SDK Manager do Android Studio. Para acessá-lo, clique em *"Configure"* na tela de boas vindas, e então selecione *"SDK Manager"*.

Selecione a aba *"SDK Platforms"* e marque o checkbox *"Show Package Details"* no canto inferior direito. Abra a seção *"Android 6.0 (Marshmallow)"*, e certifique-se de selecionar os seguintes items:

- Google APIs
- Android SDK Platform 23
- Intel x86 Atom_64 System Image
- Google APIs Intel x86 Atom_64 System Image



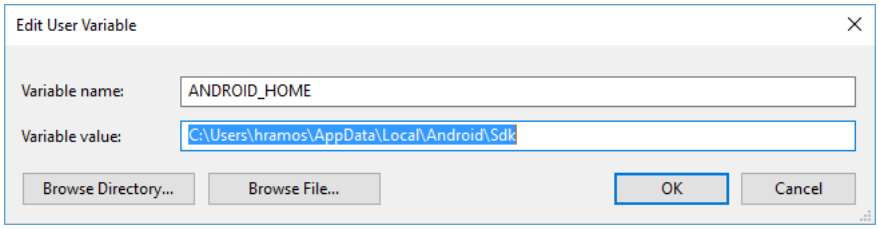
Agora selecione a aba *"SDK Tools"* e marque o checkbox *"Show Package Details"* no canto inferior direito. Abra a seção *"Android SDK Build-Tools"* e selecione a opção *"23.0.1"*.



Por fim, clique em *"Apply"* para baixar e instalar o SDK e as Build Tools.

Agora precisaremos configurar a variável de ambiente `ANDROID_HOME`, para que o ambiente do React Native consiga enxergar o SDK do Android no momento de instalar e rodar nossas apps.

Abra o painel *Sistema*, acessando *Painel de Controle > Sistema e Segurança*. Agora selecione *Configurações Avançadas do Sistema*, no canto superior esquerdo, e em seguida *Variáveis de Ambiente*.. Clique em *New* para criar a variável ANDROID_HOME para o sistema, com o caminho da pasta de instalação do Android SDK.

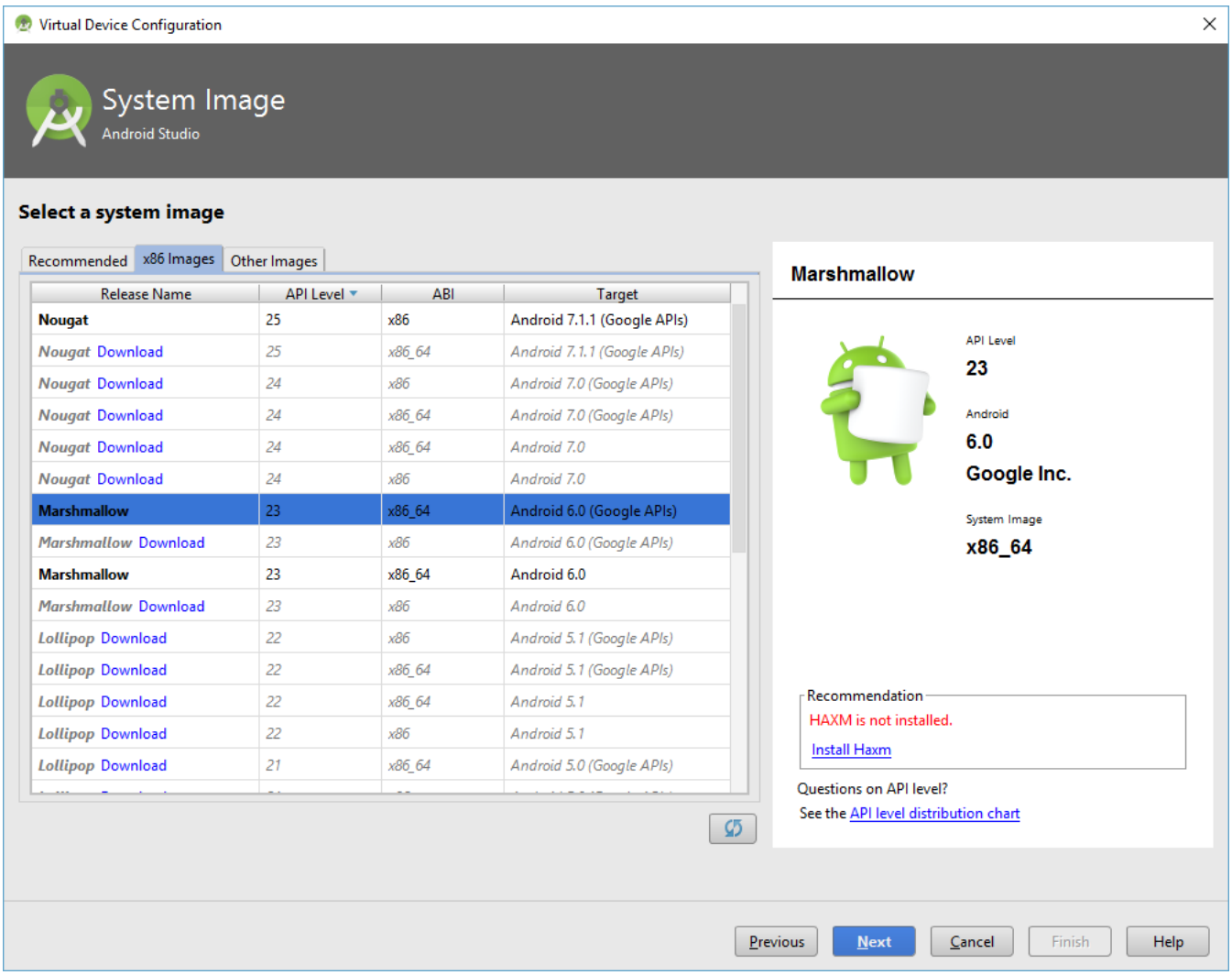


O SDK do Android, por padrão, é instalado na pasta `c:\Users\YOUR_USERNAME\AppData\Local\Android\Sdk` . Caso você não tenha instalado neste local, é possível encontrar o caminho de instalação do SDK nas preferências do Android Studio, acessando *Appearance & Behavior > System Settings > Android SDK*.

Preparando um emulador Android

Um último passo importante é que precisamos preparar um Android Virtual Device (AVD) para podermos testar nossas aplicações. Você pode ver a lista com os emuladores configurados acessando o *"AVD Manager"* do Android Studio. Procure por um ícone como o que segue na barra de ferramentas do Android Studio:

Selecione *"Create Virtual Device"*, escolha um modelo de dispositivo disponível (Nexus 5X, por exemplo) e clique em *"Next"*. Selecione a aba *"x86 Images"*, e então procure por *"Marshmallow API Level 23, x86*4 ABI image"_ com *"Android 6.0 (Google APIs)"*.



Clique em *"Next"* e *"Finish"* para criar seu AVD. Nesse ponto já é possível verificar o novo dispositivo adicionado à lista e inclusive selecionar o botão de *play* verde para emular o dispositivo.

Pronto =)