



Desenvolvimento para Dispositivos Móveis

Aula 05 - Introdução ao React Native



Material Didático do Instituto Metrópole Digital - IMD

Termo de uso

Os materiais didáticos aqui disponibilizados estão licenciados através de Creative Commons **Atribuição-SemDerivações-SemDerivados CC BY-NC-ND**. Você possui a permissão para realizar o download e compartilhar, desde que atribua os créditos do autor. Não poderá alterá-los e nem utiliza-los para fins comerciais.

Atribuição-SemDerivações-SemDerivados

CC BY-NC-ND



<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Apresentação

Nesta aula iremos iniciar os estudos do *React Native*.

Objetivos

- Conhecer o porquê de trabalhar com *react native*
- Conhecer como configurar nosso ambiente de desenvolvimento
- Conhecer como criar um projeto com *react native*

Por que React Native?

Link do video da aula: https://youtu.be/tTcl_-xNMTA

Antes de começar a utilizar o *react native* iremos conhecer as razões pelas quais trabalharemos com ele.

Problema atual

No mercado atual, temos duas grandes plataformas: *Android* e *iOS*. Cada uma dessas plataformas tem uma metodologia diferente de desenvolvimento de aplicativos. O problema é que se precisarmos criar um aplicativo para ambas as plataformas, teremos de criar dois projetos em diferentes linguagens, o que acarretaria uma dificuldade de mantê-los.

React Native

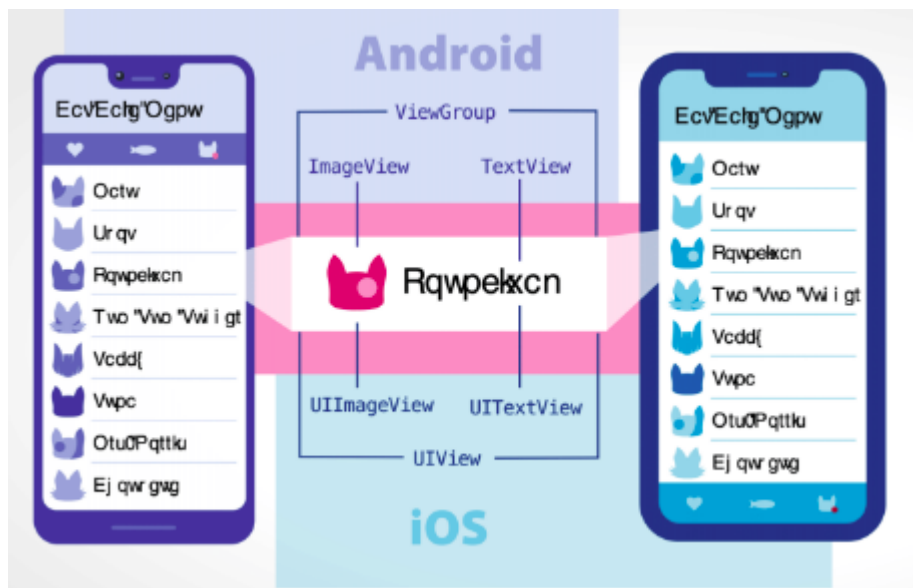
Desenvolvido pelo *Facebook*, o *React Native* é um *framework javascript* focado em desenvolvimento de aplicativos multiplataformas.

Utilizaremos esse *framework* por alguns motivos, sendo o primeiro deles o fato de podermos continuar utilizando *javascript*, sem a necessidade de aprendermos uma nova linguagem. Porém, temos outros pontos relevantes para essa escolha, são eles:

- Permite criar aplicativos para várias plataformas com mesma base de código;
- Permite utilizar *Javascript* para criar lógica de negócio e *interfaces* para dispositivos móveis;
- Aparência nativa;
- Bastante popular;
- Forte;

É importante ressaltar também que os usuários de diferentes plataformas visualizam e interagem com o aplicativo de formas diversas. Então, para resolver

esse problema, o *react native* transforma o componente desenvolvido em um componente nativo, *android* ou *iOS*.



Aparência Nativa

Preparando ambiente no Windows

Link do video da aula: https://youtu.be/tuxg6U8i_44

Vamos agora configurar nosso ambiente de desenvolvimento para podermos começar a criar aplicativos com *React Native*.

Acessando a documentação

Seguindo a [Documentação React Native](#) faremos a instalação e configuração do nosso ambiente de desenvolvimento. Nesta aula, a instalação será feita no *Windows*, mas, pode ser realizada em outros sistemas operacionais.

Primeiramente, é necessário saber que você precisará do *nodejs*, da *interface* de linha de comando *React Native*, de um *JDK* e do *Android Studio*.

Instalando o Node e o JDK

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

Para instalar o *Chocolatey* basta executar, em modo administrador, no *Prompt* de comando do *Windows* o comando abaixo:

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex
```

```
((New-Object  
System.Net.WebClient).DownloadString('https://chocolatey.org/install  
.ps1'))
```

O React Native também requer o Java SE Development Kit (JDK), que também pode ser instalado usando o Chocolatey

Então, para instalar o *nodejs* e o *JDK* execute, como administrador, no *Prompt* de comando do *Windows*, o comando abaixo:

```
choco install -y nodejs.install openjdk8
```

Se já possui o *nodejs* instalado no seu sistema, verifique se a versão do node é a 12 ou a mais recente. Caso tenha o *JDK*, a versão deve ser a 8 ou superior.

Instalando o **Android Studio**

[Faça o download e instale o Android Studio](#). Certifique no assistente de instalação do Android Studio se as caixas ao lado de todos os itens a seguir estão 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*)

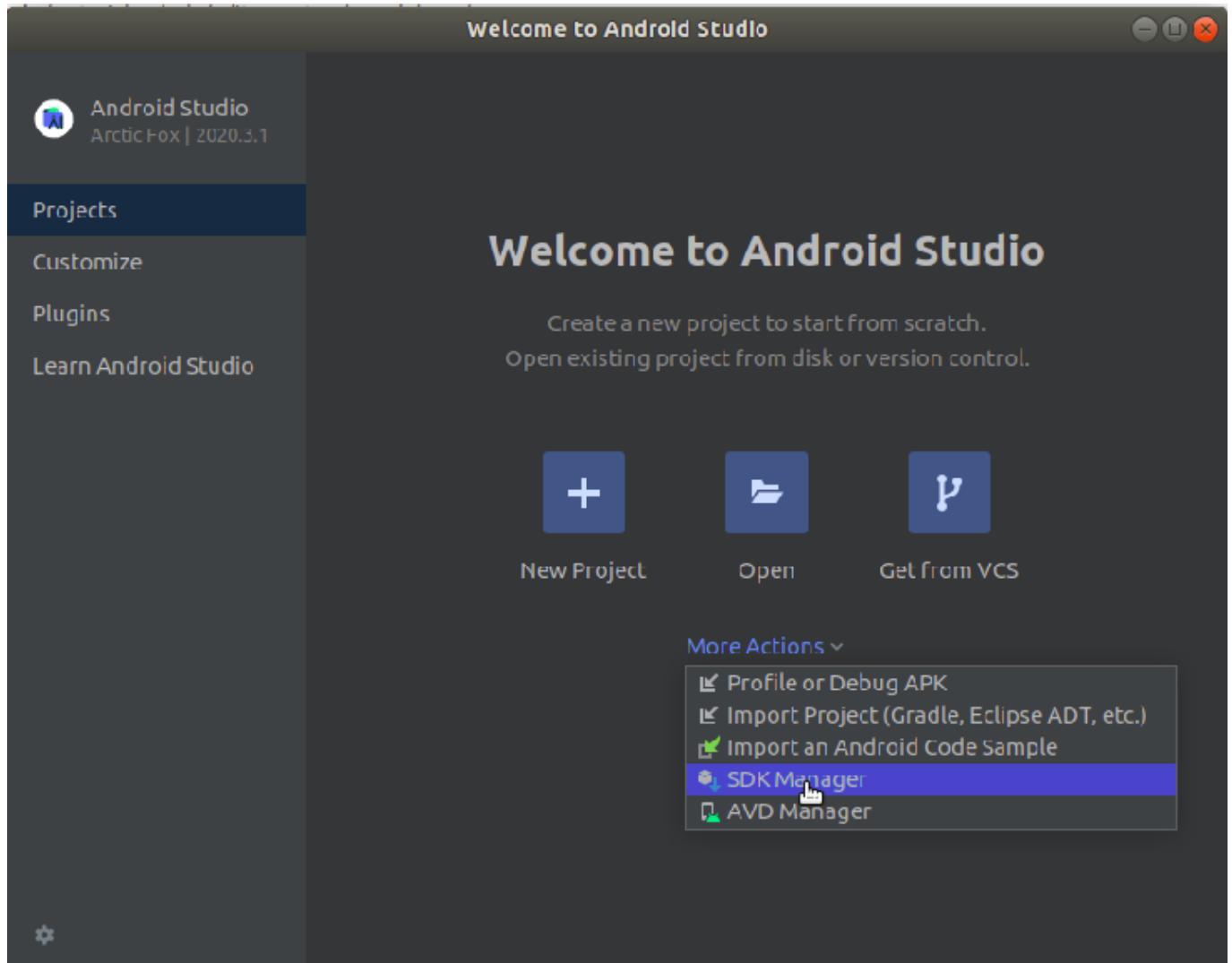
Em seguida, clique em "Avançar" para instalar todos esses componentes. Quando a configuração for finalizada e a tela de boas-vindas for exibida, prossiga para a próxima etapa.

Instalando o **Android SDK**

Por padrão, o *Android Studio* instala o *SDK* mais recente, porém, para construir um aplicativo React Native com código nativo utilizaremos *Android 10 (Q) SDK*.

SDKs Android adicionais podem ser instalados por meio do *SDK Manager* no Android Studio*.

Para fazer isso, abra o Android Studio, clique em "*More Actions*" e selecione "*SDK Manager*".



Android Studio

Selecione a guia "SDK Platforms" no *SDK Manager* e marque a opção de "Show Package Details" no canto inferior direito. Procure por *Android 10 (Q)*, *API Level 29*. Certifique-se de que os seguintes itens estejam marcados:

- *Android SDK Platform 29*
- *Intel x86 Atom_64 System Image* ou *Google APIs Intel x86 Atom System Image*

Em seguida, selecione a guia "SDK Tools" e marque a opção de "Show Package Details". Procure por "Android SDK Build-Tools" e certifique-se de que 29.0.2 está selecionada.

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

Configurando a variável de ambiente **ANDROID_HOME**

As ferramentas React Native requerem que algumas variáveis de ambiente sejam configuradas para construir aplicativos com código nativo.

1. No buscar do *Windows* digite *env* e serão retornada "Editar Variáveis de Ambiente do Sistema";
2. Clique em "Variáveis de Ambiente" no canto inferior direito da tela;
3. Nas variáveis de ambiente para usuário clique em "Novo";
4. Nomeie a variável como *ANDROID_HOME* e seu valor vai apontar para o caminho para o seu Android SDK:

Também é necessário editar a variável de ambiente nomeada de *PATH*, adicionando as ferramentas de plataforma ao caminho. Então, ainda com a mesma janela aberta:

1. Clique na variável *PATH* e selecione a opção de editar e será exibida uma nova janela;
2. Clique no botão "Novo" e em seguida clique em "Procurar";
3. Navegue até o diretório "*Platform Tools*" e adicione-o como caminho.

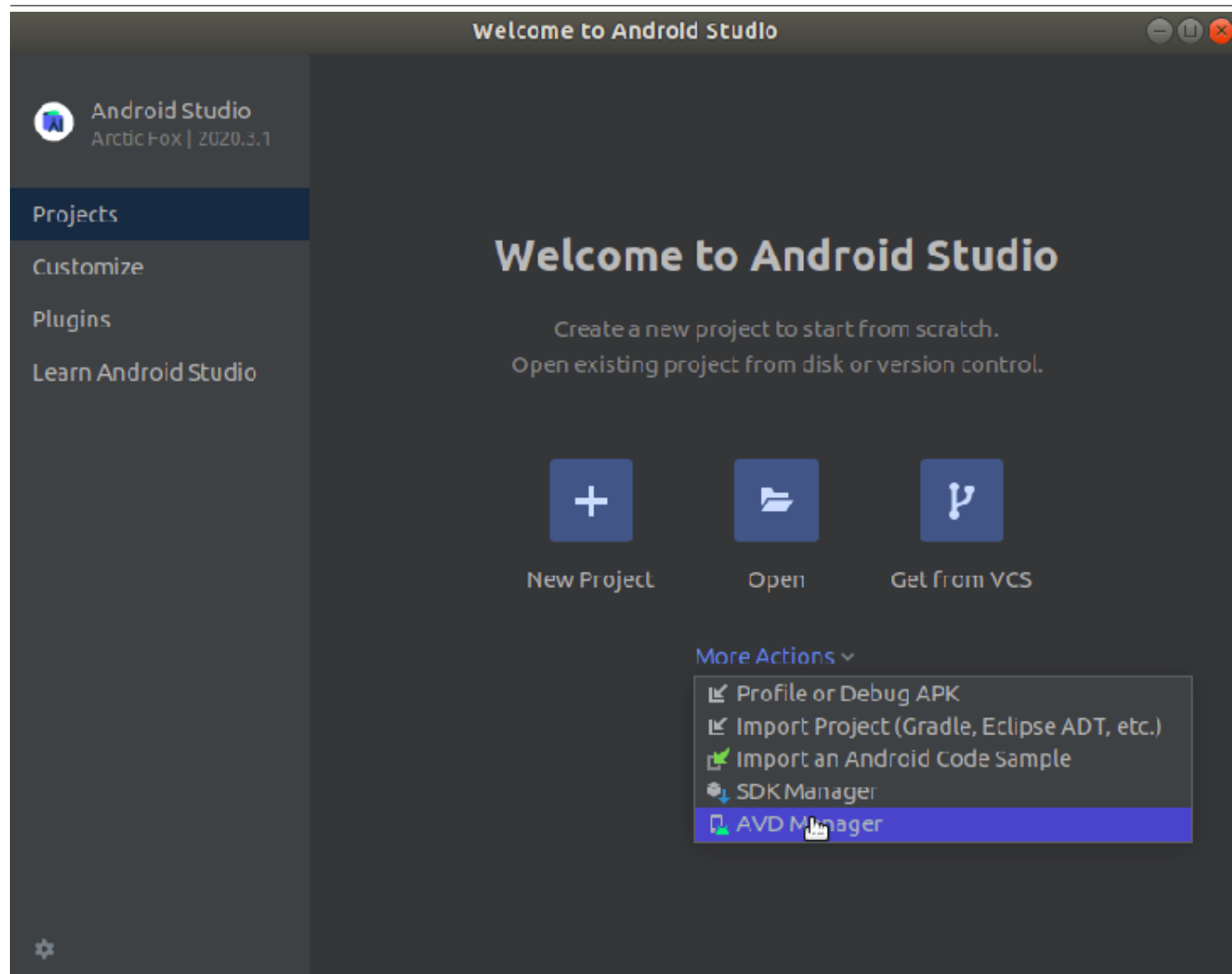
Instalando o utilitário de linha de comando

No power shell execute o comando abaixo:

```
npm install -g expo-cli
```

Criando um simulador do dispositivo

Com o *Android Studio* aberto clique em "*More Actions*" e selecione "*AVD Manager*".



Android Studio

Em seguida, clique em "Create Virtual Device" e acompanhando a aula configure o dispositivo virtual.

Se ocorreu tudo de forma correta, será exibido um celular virtual na tela e o sistema será iniciado.

Criando primeiro projeto

Link do video da aula: <https://youtu.be/AeV7vTX5NLk>

Com tudo pronto, iremos agora criar nosso primeiro projeto com *React Native*.

Criando projeto com o Expo

Abra o *vscode* e no diretório desejado crie projeto com o expo executando o seguinte comando no terminal:

```
expo init hello
```

Esse comando irá abrir um *prompt* de opções do *expo* e iremos escolher a *blank*.

```
? Choose a template: > - Use arrow-keys. Return to submit.
----- Managed workflow -----
> blank a minimal app as clean as an empty canvas
blank (TypeScript) same as blank but with TypeScript configuration
tabs (TypeScript) several example screens and tabs using react-navigation and TypeScript
----- Bare workflow -----
minimal bare and minimal, just the essentials to get you started
```

Expo

Temos alguns minutos até o processo finalizar e entrar no projeto.

Testando o aplicativo

Para rodar seu aplicativo pela primeira vez, execute, de acordo com suas condições, um dos comandos abaixo:

- npm start
- npm run android
- npm run ios # Necessita do XCode em um Mac OS
- npm run web

Resumo

Nesta aula, vimos os benefícios de trabalhar com *react native* e como preparar o ambiente de desenvolvimento para trabalhar com essa tecnologia. Ademais, vimos como criar um projeto utilizando *react native* e testamos o aplicativo.