



Desenvolvimento para Dispositivos Móveis

Aula 10 - Imagens



Material Didático do Instituto Metrôpole 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, será abordado o uso de imagens e as diferentes formas de utilizar esse elemento visual tão importante para um aplicativo.

Objetivos

- Conhecer como trabalhar com imagens padrão 2X/3X
- Conhecer como trabalhar com imagens a depender da plataforma
- Como usar imagem da internet no aplicativo
- Conhecer como codificar imagens no próprio código
- Conhecer como trabalhar com o formato SVG
- Conhecer como inserir imagens de fundo

Imagem 2X/3X

Link do video da aula: <https://youtu.be/R133HluBoh8>

Muitas vezes nós temos diferentes dispositivos, com diferentes resoluções de tela, sendo difícil escolher uma imagem que case bem em todos eles. Para resolver essa problemática o *React Native* tem alguns artifícios.

Iniciando o projeto

Para esta aula, iremos iniciar com um projeto padrão, resultado do *expo init* já utilizado antes. Então, em seu terminal, execute o comando abaixo:

```
expo init ProjectName
```

ProjectName pode ser qualquer nome de sua preferência.

Inserindo imagens

Com seu projeto criado e aberto em sua *IDE*, adicione na pasta */assets* os arquivos de imagens que serão usados durante a aula. O arquivo *.zip* com as imagens pode ser obtido [clikando aqui](#).

Component *Image*

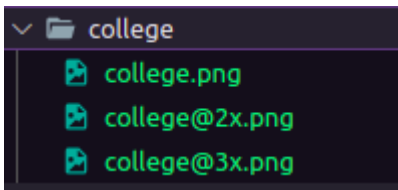
Em seu arquivo *App.js* adicione o componente ***Image***, lembrando de importá-lo do *react native*. Esse componente possui uma propriedade *source*, a qual recebe a imagem que será exibida nele. Para importar essa imagem, será utilizado o *require*.

Veja um exemplo abaixo:

```
<Image source={require('./assets/basic/imd.png')} />
```

Como já dito, existem diversos dispositivos com diferentes resoluções de tela. Portanto, uma imagem que funcione em um dispositivo pode não funcionar da mesma forma em outros.

Para resolver essa questão, o *react native* escolhe a imagem mais adequada da pasta. Veja um exemplo de pasta abaixo:



Para o *react* escolher o tamanho mais adequado é usado um padrão @2X e @3X na nomenclatura da imagem, assim, para telas menores será utilizado o 2X e para telas maiores o 3X. Caso não exista um mais adequado é utilizado o padrão.

Dito isso, acompanhe a aula, realize as alterações e implemente o código abaixo:

```
import { StatusBar } from 'expo-status-bar';
import React from 'react';
import { StyleSheet, Text, View, Image } from 'react-native';

export default function App() {
  return (
    <View style={styles.container}>
      <Image source={require('./assets/college/college.png')} />
      <StatusBar style="auto" />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: 'fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

Imagem por plataformas

Link do vídeo da aula: <https://youtu.be/g6W-DQNzrDY>

Veremos agora como controlar qual imagem será exibida conforme o dispositivo: *Android* ou *iOS*.

Variável *Platform.OS*

Essa variável vai retornar em qual plataforma o aplicativo está sendo executado naquele momento. De posse do tipo de plataforma, poderemos requerer diferentes imagens para serem exibidas.

Em seu arquivo, crie a condição abaixo:

```
const ico = Platform.OS === 'ios'
  ? require('./assets/platform/platform.ios.png')
  : require('./assets/platform/platform.android.png')
```

O valor de *ico* será o *require* de uma imagem a depender da plataforma que o aplicativo está rodando. Então, podemos utilizá-lo na propriedade *source* do componente *Image*.

Veja abaixo como realizar essa ação:

```
<Image source={ico} />
```

Com essas modificações já será possível exibir a imagem de acordo com a plataforma.

Image Network e Base64

Link do vídeo da aula: <https://youtu.be/gEKuWTPbjhY>

Agora, veremos como podemos pegar uma imagem na internet e usá-la no aplicativo. Além disso, veremos uma codificação que permite codificar a imagem no próprio código.

Utilizando imagem da internet

Para utilizar uma imagem que esteja na internet, iremos passar na propriedade

source um endereço, ao invés de uma requisição.

Porém, ainda há um problema: o react precisa conhecer o tamanho da imagem para exibi-la. Então, adicionaremos um atributo *style*, informando a largura e a altura que a imagem deve ter.

Veja abaixo o exemplo:

```
<Image style={{ width: '95%', height: '70%' }} source={{ uri:
'http://reactjs.org/logo-og.png' }} />
```

Codificando imagem no código

Para codificar uma imagem iremos abrir um novo terminal e acessar a pasta */assets/react*. Dentro dessa pasta existe a logo do *react*.

Execute em seu novo terminal o comando abaixo:

```
base64 logo-og.png
```

Esse comando retornará uma sequência de caracteres que é a codificação da imagem.

Copie essa codificação e crie a variável abaixo:

```
const data = 'data:image/png;base64, {CODIFICAÇÃO DA IMAGEM}'
```

Agora poderemos usar essa variável na propriedade *source* do componente.

Ao final da aula, seu código deve ficar como se observa abaixo:

```
import { StatusBar } from 'expo-status-bar';
import React from 'react';
import { StyleSheet, Text, View, Image, Platform } from 'react-native';

export default function App() {

  const data =
  'data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAABLA AAAJ2CAMAAAB4notuA
AAC7lBMVEUiIiInMDMwTFM2XGc8b30/eIdBfIXEhJZBe4w/doY7anc1WGI tQUclKy0jJ
icyUlpCf5BQqcFdzexh2vtcz0tQqMBEhZc4YW0p0Dw3YWwjJCU0V2FLm7Bdz+5Ywd1Ik
KQ3YGsLLC5YwNxMm7EjJSUmLzFbx+Vf1fV0pLs6aHUnMDI lKyxg1vdNoLZe0082W2Vax
+QoNDc2XGZMnbRg1vZKlaouRUtMnrQvSE9ax0ExT1dQqsIvSVBRqsNAeYlh2fouRUxAe
```

opBfY1h2f1LmK5Lma5Cfo88bntf1PNDgZI+dYQ1WmRg2PhaxeI3Xmk3X2krPUJPP78rP
EFPpr5e0vIkKStWutVf1PRXvNhGip1Zx0FNoLdFiJs7bHlGi55XvtpVttAtQ0k/doVFi
JowS1NRrMU7a3gpNjokJygwSlI+c4E9cX9SrcZMnLItQkcyUVlFh5laxuMsQEUjJiY8b
nwxTlYiIyQzU1xJlalF0/IiIyMxTVVYv9w+dIJcyuhNn7UmLjBZwt9PpbwoMzYmLS9Hj
6I2XWhUtdBbyeczVF1TsstCgJFdzu1XvNdRq8QnMTRQp79Yv9tby0Y1WWNIkKNKlqsLK
isr00A4Y29cz0pTsMonMjUq0z83X2pJlKgtQkhJk6grPUFe0fA0V2BLmq90ork9cH5Pp
b1Gi50kKClSrsdWu9YpNzssP0Q5ZHBcy+kuRk1TSCO5ZXFFiZxAeYgq0j5Sr8hg1/hUt
M9KmK1Us81Dg5Q6aXZCgZE8bXtWutZDg5U+dIMwSlFVuNNWudRRrcVNobg0VV44ZG8yT
lgyUFhEhpcpNTlAe4s0Vl8oNTg6Z3QvR00xTVRvt9JZw+AuREp0o7osPKNJk6c9coA/d
4ZikaVMnbNZwd5Kl6xXvdLgJJ9Bfo4kKCpVt9E7bHo5ZnJe0fFEhpgzUlsq0DxTssxIk
qYzVV44Ym45Z3NUtM5HjqJHjqEq0T1DgpMvSE4sQEY9cYBSr8k6aXVHjaDcQTkTAAAmz
0lEQVR42uzBw2EFABQAsFeb37a9/1i917adJAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AA
AA
AAAAIAfBwAA
ABgLi7A3E0H8Z22jm6KawDb07t7+4mbkql0JhvfIJtLp5KJm/b3dne24wRAvLasJR5Ur
lRr8YVq1Uo98aBSO9AM4L9rLbUTT+h0e/34Ev1et5N4QnupFcB/NlgtJZ4zHBXy8cnyh
dEw8Zzx8iCaf2syTbxIe3cWn2i20z1mD54Z5gyiMIDeMtaNVX22vbZt2/z/SBetZ95wn
nN4KUcnBABq2jnlpZ2dX9AvcnF+xku73CEAUNDVNa/k1Q39Aje3vJLr0wIA5Wxv8qp0N
6SxGx2van0bAEAxegMLuDWShoy3LMBgIgBQipnFWKxvSSNvbRYWYyeFAICDhTldu6SBX
ZeThbkJAJTh8bIEn5+k+X0swRAgAFBE8BPLCYVJSjjCcQKPSAkAEIuzrIR+jYSt6RMsk
xkjFQBAiqdLZ7L3czm30b9V00NFiiUSVCryImeFrbY5nMuVK5efeboUAYACql6eZKjVG
/RNs9U+/cRz7T17QgKePNvjuT6dtltN+qbR6Rp4kqVKAPDfW0vyBE0vTxPWBsMRz1Mc0
8rGRZ7naPh0jSY86hn4C3v3/2bVuL8B/BbgHAbdBcNI06SYGmKUKB07xGyQUUQE1VA4L
YbZ0pnMqJGpoU0pSiCMJIiS0BABLiojEdAJCDhffvtcHHx6nr3W2nuvtXbVmvv1BzxV1
7Xf13rez/tLir0GQ0Si7kqmKBwBZ5VXXV10d1XbJJGV5DZVdFd+2FWVcHbNSKa4FiIRJ
61H0fbXneGu+sLRdHddDbJQcx3djb6wGu6Ku9DWvBYi0tRK3K9PwLP+mLF0VXcDMnZFH
V2NHZMPT8nrm1rBu4gUN6elVRJp/Xkc3cRvzEdGKrdL0M24PyeRTrIVLfxFEJEoG0xLt
zxkomw83UyoQQYabqKb8WXIxPButAyGiERY8maa6hqQmaI/TaSL/S9GWpccTBcT/1SEz
GxVR9PIJEQkuo6gpQwZK969lM7aDUQaf2pHZ6W7FyNjF6PlCIhIdB1G0znIxraJ6eKWP
HjIu4Uuug1DNs6h6VaISGRNKqehw2Rk57YpdDa1Bq6mTaWzKbch09NLaSifBBGJqhk03
Y5sTWtDZ11nwsXMrnS2zzRk6w6a9kNUichxNMTvRPY0qa0ju+6Go7vvoq06Q5C9XnEa7
kFEiUhRAQ1nwI9ps+gocS8c3Jugo1kN80M+GgqKEE0i0oymreHPWHI6uj8flvzz6Kh8I
PzZmqZmEGkS1JZTUAGfThtLR10aYWjsQkdj74RPefU0HItoEpEjaTgMvLwCHa0TB6qXg
eoH6CR2dkVohRlHIpJEJL+chlMQwIwS0pn9IP6w53g6KtkFAYyhoTwfUSQiw2hIVC0IO
8fTydA5+M2coXQy+04EUZugYRiiSEQeomEcFLFHVNmPoxfPTyFTroUI5hHaHgUUSQij
9FwEQJKzkvQQcERAHBECR0k5iUR0EU0PIYoEpGWN0yFwPYuoIOqvYE/V9FBwd4I7NSmk
HUXkb409EJw042mg9JTTimlg9E7IbjHaTgYESQijXFuqLwSIIdjqCTqYP580nmhACCrLa
ZiE6BGry2lYgFBUPMkM9ahAKBbSMBPRIyJ303AcwpF8ihl5Kolw9KBhb0SPiCy4WmE5
ZkE00o8g7A8TcMiRJ6IOgkHIjRlVUyj6iCE5k/qJow+ketpKEN4mtXTU30zhKeMhlsQP
SLShYajEKIL6uih7gKE6CgaDkX0iEg3GqYjTJNH09XoyQjTdBpaInpEpD8N1QhVp/F0M
b4TQlVNw0JEj4jcTEMRwtXwdzrq3YBwFdfwPkQkerpyQ30QtobZuYpXlp7cUFeISPSYy
+ILELbJQ+lgaC+ErYAb2h8iEj0FuQ1YNYV0VNgp1/8QEYl6wBqKcNX2povetQhX18gHL
BEpyGXqJ68bXXXLU8ASKc0n9ZPsQQ89kgjT0MgHLBFpnsPf+bP09Gw0I289RCRCcn+Te

o5pXJjLu62IRD1gJRCesgTTSJQhPPMjH7BE5PxcVbo/X8W0qp7PVaX7zRCR6HmBhgaEZ
NBEZqBuEELSQMNiiEj0vEjDaTksWkqqymE51p00vAgRiZ5zcjIPK78lU8wvK+vAFLv15
2Qe1ksQkei5moYlCMX9TBF7GXg5xhTnIRRLaLgaIhI959KwB8IwkKnuBYB7maofwrBhk
9tVL6ILFJciBK+UMsWzrrWkpa8iBJdGfwmFiGxDw2sIbvpcpz4c926duZMR3Gv295yIR
M/WVmBBYDv3dux09uiH7t069EWqW0NEoudwGgYggQX5+IFr4KHvyYR1AAaDoeIRM/rN
IxEUeczxVCjPHTQUKZ4A0GNp0F1iEj0FCW4oSoEtCRBW9XDMHSvoi3xZrgj3RP5EJEI2
p+GTghkcj1t8TJYyuK01U9GIG9RI92bAJGFNIxAEGcuZop5SDGPKRafiSBGNIW1hCJyD
w1LA7/VWw6Hg9uZ4jgEcQAN90BEouhpGp5CAM8wxYt5cJD3Il0cgACEouFpiEgUPUpDG
/i3rJS2vg1w1NCXttJl808yGh6FiETR2zQsh2/VK2irWgkXK6toW9EWvi2noRlEJIqm0
dAzHz4l36Etth9c7RejrX0SPuX3pKEBIhJJJTS8C5/mMcWJ8HAiU8yDT+/SUIImQUR1D
afCn7fn03ZdJTxUtqctcQ380VVVDU2DyD9o2BW+t01KW2EtPNUW0ta1LXzZLYZ/QESi6
T0a0sKP5Eu0VV2ANC6oou2lJPzoSMN7EJFo2oWGQvhxL1PchrRuY4rB8K0Qhl0gItHUQ
E08Ftl7uB1tZyMDZ9NW+jCyVxvXI2ETIVIX+P0keCRtT+QhA3lP0Day0PBHYh1EJKpa0
rCtr7y9Ze7ryMiDLWj7B7K2iobdICJR9RQNhYJbY5jixbPfXz1rtyPHjj3/g6H1v+Bv6
n8x9IPzx449crdZq98/+wmm0AXZ0tReeCEiUTWGhr7Izs4PN2eomj+8M7LTl4YxEJGoe
pCmachEzVEHDX7qw25rypkD5Wu6ffjU4I00qvHTXcQH0VSIK0teBi+tHz7lpI/eGd+TG
0XP8e98dNIpD7eGl7Kmk3MXkXdo2B208uYcsuP7L9Zxk6h78f0dD5mTl9F0wnCGItH1M
Q2PwFI059SPD10zn5vc/DWHfnzqnCJYHtHW56ZDZAKnpZ/gD61fvfTJF0q5WSld/uSlr
7bGH84spWEJRCS6WidoeBW/qF4yb/W+MW6mYvuunrekGr94lYZEa4hIhH1Kww75K5857
KwYN3uxsw77bGX+djR8ChGJsvNoqC/nFqS8nob7ISJRNoMRMgMiEmWdGCGdICKRtoKRs
QIiEl15N5w3mhEy+v4b8iAiEbTV54eWM0eq6jdQxZwr0ftZrSAikdLw2W4JhineonBBY
8ta7X78qn4vb33V88t0m/56dTUcVFe/Pv20Zc9ftfXL/VYdv3urylouKGwRZ5gSu33WA
BGJiGnPdEswVC1rEUjt6QxVotsz0yAiW7xJL9+UYMjqByGgQfUMWeL0l8+EiGzBKq+6e
i19S+x7xnnvHbTyNdo+R2Cf0/bPlQddet7p+ybo29qr11VCRLZMXzx9Mv2pf+CWC8vez
cMvTutDyzkIQUda+pz222ibsgtbPVFPf7o+vSdEZIUtX9Y+zuyt/fKra294C/8v0YGW5
jUIQU1zWl5M4v+9/vVz/1i4ltlLvFOWDxHZknQ69mRmq/mRu38zpxKWRbQdgFAspe1Ps
FQ0+2b3I5szWycf2wkisqXo/GEps1I+ZPet98zwQ6hlEqFItqSlvgZ09hyz+5ByZqX01
s4QkS1AcsluzMrJn60sgpsnaSmfjpaA8Xk5LD7gpWvnMyczKbkuSEJHNW9HS8czS2ka4e
jVGy7UIzbW0xK6Aq8a1zNLspUUQkc05XBUye2VwM/wsWhbmIzT5X9KypgJuypi9QoUsk
c1W0beF90NquJlHS4eZCNHMUlq0gZur6Ufht0UQkc1P8ptCZiK+focjaWg+HM6+qKLlR
oTqRlqqHoez4c1p0HKH9XFmovCbJERkM3PJ0Gag/JxFNCBtNF0FZ7NoGV+BUFWMp+Ud0
FtH08tAzaJzypmBcZdARDYnvVbHmFbJkwc14he17Wj4Do7+TEu8GUJ2cZyWg+DoexpKa
/GLxo0eLGFasUN7QUQ2F588Vcp06j8sq8DvzqBhVBECVIyk5TyE7jxaRjfcQf4oGs7A7
yrKPqxn0qXPfgIR2Sx8/QHTaPfSiCZlbiBNN2SUct+5GKEr7krLdnBwA00Djch6yEvtm
MYHS7DpiUjNPuzjkw3awjQtnv6d8PW1tIxBDoyhZe3rSHU7DfEamNpu8wjTuKcGm5iIv
FxPT6NeexephtBQX4EUx9FyHXLi0lq0Q4qKehqGINW7r42ip/qHsCmJSHUbeo1N2KMCT
gbTtDVsi2I09Zm0nJjeh6bYCNgoWkwnFTsMSFGL5dVY5MRkXUn08NdH70LFzUJGtrAk
r+c1h2QIzvQsjwfljY0JGrg4t2P7qKHk9dh0xCR/03jdNf3pGq405KGqp1h6kfLzRXIk
YqbaekHU3E5DUfCXfvJfekuvns+NgER0XA3uht7aj68LKJpKQyt62j5GjnzNS2jWsPwL
U2L4CV/xli62+1AbHQi0n0FXU29G2m0LaVhAAzP0nIOcggLLdvDMICG0rZI4+6pdLWi0
zYyEdmjJ90MeBXptach9gM28EMfmvo8jhzas4qmPubfJm6376T3ygC66bkHNioRuTRGF
20/RiZe9qrXb0076bnyhyWDT3zsq68e03Hwkh8qkaHtaGnjLZW/DZn4eidxC7FxiMiy
e3pYuR+SWTETmTvm8QfLo7R1PVMZ0Ld92YVcAMls957F5loHE1TrBn+kCykobwYGUNuV
0gX2yexkYhI/tV0VnBSHjL1IU1v43fJL2mZgfQ0HDyWDsY0PhDpnULLl0n87m2aPkSm8

k4qoL0r87FRiEj+k3QUu6MTMreOpq/wu71oGYC0av65li7W/rMGaU2gZS/87iua1iFzn
e6I01GPfGwEIpJsRUeLL0Y2Kk+mYW0x/mf4vjTFuy0N4hN70kPPE4uRRvc4Tfs0//3st
TScXIlsXLyYjt5PIvdE5Fw6affGcGTnR5dyzXtpuR1pL0nKNLq+iTTuoGWwSwrnj8h00
THt60Rc5JyIzK0T/pcjW3NoWoBf7TyKprVvwdPwf8aYVuwvw+Gp01qaRu2MX62naQ6yt
VN/0pmHHB0RGTGmip1bgeyNo6k7fnE8LcfCU6dxzMi4Tlm2F06KX3Sm6Utkr+Jcx/+0G
cgpEZLzXVSj7oYfJ9BhUnLNWpp0PhNe5nRlhr0gZcz7YPW1gDAdzSdAD/uHsVUVTORQ
yLSeiRTLrgEX2rLaSiYB0B7Wh6Fl5mjmlFRM+HLIVq+B3BmAQ1VtfdL9QVMNbI1ckdEP
mSqJxvh0+00HQBM7kDTgiQ8DBrKLAX9EB6S62nqcCdwAE23w6fGHkz1IXJGRGYwRexo+
PY2TZ8CPWjZBR4aCpmVwgZ42IWWHkD/1PJWv46JMcV+yBER0XAUBYmBC0DvNF2yU5ym0
+EhvyWz1DI fHu6jKb7TJTNRgADE7SN0hC5ISJf0VY6A0FcSLOPLjTF08PDdszax/CwM
k5TlydpuhRBzCil7SvkhIj8FKclMQ0Bt01DQ4dYyqXMw8PzmbX5D2eToot1oKHPgQjk1
AQ8Z+QCyJyJC2xgQjo0Hpp9wPcVX5KHz7Nh7sv2tHTcQhoYIyWI5EDIVIqbUcj qBH09
Bg8LKivi+DhInq6BkEdTdurCJ+InE7LPkkE1pseSraCu8aD6cvBjXB3YAE9LEdgyX1o0
R2hE5GfYzStKUZWJ9DD8fBwJX06AR4+poeBCG7SGppiPyNsInI/TR0eRgiK76Kr5rVwL
xxJn0Ym4a51c7qqn4QQPNyBpvsRMhEZ3pymHxGK8+jqa0+iU9/ehodVPjNq/sfqNB+0c
InImzQNPR0heDdGF6MmwUMr+tYKHs6so4tYL4Qhtcv6TYRLRK6n6YTwc/nZFGk0r6dv9
cPh4V+0hZ0dX0rTLQiXiHxAQ9c8hGSJn9c8XMMARsBD41A6W4KQFI2m4Q0ESkTeomlbh
CU53sdjHo5lAMfCSz86Gp/M2dDwt5BDIkphxR9EaAbSydA8eLmPAVwHL3l9w65psD0YV
xIrl0R2pGEIwvPJFDq4Ep5WMIDRPmrDpnyC8AyhYUfkkIgm9+2KED3NVPs3wktjnAHEv
Q+v0JipnkaIdtUcvlwSeZGGuxGiTqVM8S94msNA5sDTYKY07YQQ3U3DAwiTiJxPww8I0
6201X0CT9cwklfh6Z0JtN2KMH1Bw80Ik4h0pWESwtQ9RstJaT9RANKt3q6lrTvCNImGr
giTiNSbFySEawJNJWfC2wwGMgPeziyhaQLC1Y4bqkeYRGQoDUVATt8Jf4K3MgZSBm8/x
cJ9I7Tl03AywiQis2mYhjANpG0WvF3BQK6At1m0bYmWnDAwG2ESkak0dEaIKs+nLdYZn
n5iID/B08oYbYWVCFfnGqYiTCIyi4bPEKKDm0qv8PQ6A3kLnrow1UEI0Wf292SYR0Q1G
q5GiAYwVwmpJUwgBJ4mhljqhcRoqtpeAlhEpFHaagrQmi600kbeFrAANbDUxs66Y7QF
NXR8CjCJCKdczd07LY6if8ML/9gAP+Al2Fx0rk1Z4Xu7IwwiUhFTxpWh96YY7sMXr5LA
N/6+MAKszlnNQ09KxAqEdmNhviwsFufbbGV8PBWjL7F3sryiTDk9ufp82nYDeESkwtp2
gfhyJtIF7PgZT19W5DZe6itLg/h6EHTtQiXiPwQoyF+CUKxB93EnoeH5+jbSfBwVIxuv
kEoLonTEPsBIR0RqTQtzkMY/k1XN8HDtFL6VDRn106Ms0o785aHfqxNRA6g5cfQXx9tR
8DDh/TpVng4gh66IwQn0vIf2EQk9A7lxNcI7n16GJIm0e5L7AJ4GEIP7y04GxI56Ku2i
ch2tMzthaCqq+hLb+/iAF9W+5+z1bMaQfWaS8s0CJ+ItJ5Ly5oGBPQcPS2uhLvp7ehDu
+lwV/kCPT2HgBrW0DKqNXJARAbTtr4WgSTX0Nvn8HA8fTgeHj6nt/OTCKR2PW33IhdEZ
HghbRMmIYgRNI08mKa+jXBXsZhZW1zhdWBfmvYfSdMIBDFpAm2Fw5ETI1LGF19Uh7g8j
I0vz0oStlMVslS1U1YX1G0G0/QkAqh+hCnKkCMispopFtfAt+qeNKytzltBU/NaENGjx
qzE9oCH2hY09c2rXhte2r1msccLQ0hEZFodU6y4AH7dS9P1wH9oeRZe5jEr8+DlWVoWA
dfTNBh+XbCCKeqmIWdE504YU6w9BT71pukCo0gsmqr2hJePmIX/wsueVTSdVQRcQFNv+
HTKWqaI3Y0cEpGnmCr2dD78eJ6mLwHgEGaVNkr+N4t4lYSXJ2nZGgC+p0l5+JH/dIypn
kUuiUh+SzqY0Ak+tKJpkWPPYuwoeHojxozEdoCnw+1zHgGQekdtBR86TaCD3fKRUYJSu
4Y0WoxB1hrrabir2HkL/b+T8DSmhBkoGQNPYX/Tcg1+UWydXtCIrI2ZQgfn1yLHRKTXK
Do5rgFZ2oum690GUu0Hb5MfYVrjJsPbDLdxXNfTtBey1HAcnYzqhZwTkZkFdDL3NmTn0
pqW4X9+TtA0uhHeKt+rp6f69yrhrWI0TYmf8T/LaLoP2bltLp0UzMRGICIXl9DRdb2Qh
Zr5NPR3XzCxCuIs9VgVXVU9thXSWUXLHfjdWBrm1yALva6jo5Jm2ChE500S0irdvhgZ0
8l1GfygKppKapDWtKda0NGUp6YhrZoSmqoG4XdXZjGw1FK8fSkdlbyNjUREfmpBZx0/r
USGXqChtK1HV/NhyEDFKavraalffUoFMnAYLbviD23b0bAYGar8diKdtfgJG42ID0tLF

73LkJGVNHXB/5u0P02xS5CRymXP3b08gL8qWH7PScsqkZFLYjTtP8ljc3lnZKSsN110H
YaNSEQ6fUo3U0cgA896dQEvPWBJTLX0Gj6ntWvYfZlAlqWGPHT73niKl082knBFQiM
mkfuup2BNJJjZhlHAjgvSn5U/IoT/R0t8Id8NH0bAiiXS06EZXBqYkxYLI8sIEXT2xd
xKeLqHpMRheoWVuNXKmei4tr8DwGE0Xw1Ny7wfoKnFhEhufiKybQne9FzXCw9lpFtJ0p
0U85Mz9tHRMs9rnbHhoXNSb7qaswyYhIns0oIe5N9bATeVEGmbDcmc7mhLdkS0dEzS16
wXLbBrqKuGm5sa59DBgT2wiILK5qh09LB53BJztKnZ1zL00fFmJnKgcl36Iwhs07QJnR
xzXjh7ararEpiMinWfT05pL28LBLTT1gq34YFo+Q070o+XgYth6ZTKyoe2la+hpfGdsU
iLSuH0HeurT4818WCrraFiAVLfR0qIB0dDQgpbBkGohDaMqYcl/s0cfeuwqfSM2NRGZ+
W+msf9fZnovhH80qZIP0HI7cuB2Wh5IIItVz3kv0Z160P9P490yIyGagsl8901m86nH8v
3NpiD+YUTI8NgKhGxGjKdEZDl6Pu1dhPL5qMd0p/6wSiRj5qLkjzrTGbjSzv1lBwxA4+
o6WsyOQsoqzaPk0jl6kYQV+M3nbsUwrfkcNRGTzccEZzED/Y1YCwM0ZraI5sAutxyJkx
9LS4kA4utdpeNfKYz5lBs5YCRHZvLw5npno+v2bjU/TEBuU6QNeu2EI1Zx2tAyEs9djN
Dzd+0b3XZmJ8W9CRDY7RUsLmZHyAhoWwkXl17Q8kUSIUvP6j7ieP46GgnJmZN+lRRDZP
Clk7UsfVsHNsgQtJyBEz9AyvzPc7EiLwlWTJApZd2axJ/WuQQjNoBJazoWr0xWuokckf
8ZCZucsKv5b7Ejr7aDl4Z7g7i9lZ0CMfIrLZu+avcWZjYsdtXy2Gs/loW4qQLKVtB
pwVv7rt0R0Zjfhf38aWQUR6fV/C7CReu0PeZsVI1Z6W+rcQipp6WtojVXGzWxcsTjA7J
d/3wpZDRIOh9mfW4mvuWfX1IBj2XEtLR4SiIy1rv4Dhwa9X3XN+nFnRP7AYWxgR0fwf5
fSjYGqrf63rhN+9R9seCMEetL2H33Va969WUwvox9qvfsKWSERq+w2J0ae7+u9z40NHV
QP5C2lpXoPAaprTsJ4faHvUQ0+36X8XfYoN6VeLLZaITN+ukEHUf9rxDNraI7B3aDuj4
6f1DKJwu+nYwonIxR9NZMj+g4D+w5BN/0hiRIGIVB5x3v4M013nbnfvy3tfPKymGFkpr
hl28d4v37vduSUM0/7njahEZIhI5Yj7+zIXCiZ+MPaBlqtX397qo+233/YXl/b71aXb/
mL77T9qdfvq1S0fGPvBxALmQt+zFa0iSKTzx+vjjJT4+o87I6JEp0aAUyYMKQfUIMJEZ
HgVI6MqD1EmIs8zQi5BlInISYyQCxFliInI0Dde90b45txDN27/RnoaXEGEikpxCw+dA8
rSH7h9Xxc1a1bj7HzotCXx0w5QkoktEhtE0Hf+Tf/lDjw0o4WaoZMBjD12ej/95nKafE
V0isoigG7Gh50RTtrtsdgduJjrMvmy7UyYnsaGDaviE6BKRvjS0Qaq8mXuduHp5FTehq
uWxnbjXzDyakPD+4guEelPwzZw9eAuz1z0zuxyblTls9+56JldHoSrbWjoJ8gSkcb5N
HRH0g2Hz9jx+vaLpzCnpixu/920Mw5vQDqdaZjfiCZCRGWjPYuQqcbJI16+8LFZDNmsx
y68bcTkRmSqqKqplI6KyGAaHgk2iD24jsjWVBr+hagSkVtp0BvZ0vBg2jq2nb7TsmZX3
TBjxtJ+/S7c9lc33rjtry7s12/pjBk3XNVs2U7T275D2/4HI1tn03ArokPEetPwELK2L
kbbQGTkM9pi65C1h2jojYgSkeEdaHgX2dudtj7LkIFlPwL7Ddl7l4Y0wxFNIjKThpIks
jd8HG19t0JaW/WlbVwespcsoWEmokLEbqPhCfjxQz1tN+UjjfybaKv/AX48QcNtiCYRe
ZaG7+DLqUzxI9L4kSl0hS/f0fAsokLEZtFwAvz5nrbYGHg6JEbbd/DnBBraI5pEZDQN1
8CfxhdoK/gZHn4uo02FRvhzDQ0rEEkikpegoRY+nVZCW2E1XFUX0lZyGnyqpSGehyZAR
MowusK3U5ni9Hy4yD89UALLdrBGYjUFIIn+jYtF4dzZT/AUu/sIUj8G/I2n4G6JIRC6l4
Xr4lzeVKR6Fo0eZYmoe/LuehmsRRSLyfYg7Zx6cS1uf5+Hg+Xa0zX0QAVxIw/eIIhE5g
4YyBLEkQdvch5Dii7m0Jb5GEGU0nI4oEpHzadgJgWzLFL1rYantzRTbIpCdaLgZUSQia
7mhWCMCSR7KFNflw5Dfnik0TSKQxjg3tBYRJCKtaahDQMwzmeK7tI+Js4sR0EQaWiN6R
GQ0DWMR1J0FTHGt1UZjK7gTQX1JwxxEj4hcRcMsBLZ3grZEGf5QlqAt/mcE1oWgdYgeE
fnWLSMKbh5TVD2P3zxfxRTzENzZNHyL6BGRHWnYASHowRRze+FXveYyRY8kgjuGhh0RP
SLyFA0HIASNXzJFYQMANBQyxZeNCMEBNDyF6BGRW2g4BGHoNJQpFtcCtS8wxdB0CMMhN
Nyc6BGRNjS8glAsq2KKbnl53ZiahlC8QoNbRA9InITDSsRjrIYU5zTkSliByEcK2m4C
dEjIgtPGISQzGNG5iEkG2hYiMgT0RbVYoTl02bg04SlWLtUo0+kLw0IzG0kqMcms0Bo6
IvoEZGu3FA5wmPPZPcc4xBcuTXp0XpEpI4bKkCIBg2lp6GDEKICq4k7SkQk97/zmQX0U
DATYZprHo7oEZGCXN6krqiiq567IFRdFbD8E1HAskcz2AMcFLBEZLNK/ZxAFyfk0hkXP

[illegible]

```
return (
  <View style={styles.container}>
    <Image style={{ width: '95%', height: 400 }} source={{ uri:
data }} />
    <StatusBar style="auto" />
  </View>
);
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
```

```
    backgroundColor: '#fff',  
    alignItems: 'center',  
    justifyContent: 'center',  
  },  
});
```

Background Image

Link do video da aula: <https://youtu.be/SgvgPsAc8IY>

Continuando o assunto de imagens, vamos ver agora como colocar uma imagem de fundo no seu aplicativo.

Inserindo imagem de fundo

Para inserir uma imagem de fundo, utilizaremos um código simples que pode ser copiado abaixo:

```
import React from 'react';  
import { StyleSheet, Text, View } from 'react-native';  
  
export default function App() {  
  return (  
    <View style={styles.container}>  
      <Text style={styles.grande}>Olá</Text>  
    </View>  
  );  
}  
  
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    backgroundColor: '#fff',  
    alignItems: 'center',  
    justifyContent: 'center',  
  },  
  grande: {  
    fontSize: 50  
  }  
});
```

De posse desse código, adicionaremos uma imagem de fundo que pode ser

encontrada na pasta `/assets/background`.

Para inserir a imagem no fundo, o *react native* possui um componente denominado de **ImageBackground** que irá substituir o componente **View** no código. Esse componente possui a propriedade *source* que recebe o caminho da imagem.

Após as modificações, seu código deve ficar como se vê abaixo:

```
import React from 'react';
import { ImageBackground, StyleSheet, Text, View } from 'react-native';

export default function App() {
  return (
    <ImageBackground source={require('./assets/background/bg.jpg')}
    style={styles.container}>
      <Text style={styles.grande}>Olá</Text>
    </ImageBackground>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  grande: {
    fontSize: 50
  }
});
```

SVG

Link do video da aula: https://youtu.be/_Yv3nIGFW4

Nesta aula veremos como utilizar uma imagem vetorial, que se adapta a tamanhos diferentes sem perda de qualidade. Iremos trabalhar com o formato SVG, muito utilizado na internet.

Adicionando bibliotecas

O *react native* não dá suporte ao SVG, então, precisamos instalar bibliotecas que auxiliem nessa questão.

Para instalar as bibliotecas, execute em seu terminal os seguintes comandos:

```
npm install react-native-svg --save
```

e

```
npm install react-native-svg-transformer --save
```

Configure o projeto

Na raiz do projeto, crie um arquivo nomeado de *metro.config.js* e insira nele o seguinte conteúdo:

```
const { getDefaultConfig } = require("expo/metro-config");

module.exports = (async () => {
  const {
    resolver: { sourceExts, assetExts }
  } = await getDefaultConfig(__dirname);
  return {
    transformer: {
      babelTransformerPath: require.resolve("react-native-svg-
transformer")
    },
    resolver: {
      assetExts: assetExts.filter(ext => ext !== "svg"),
      sourceExts: [...sourceExts, "svg"]
    }
  };
})();
```

Se você estiver usando o *Expo*, também precisa adicionar isso a *app.json*:

```
" packagerOpts " : {
  " config " : " metro.config.js " ,
  " sourceExts " : [
    " expo.ts " ,
    " expo.tsx " ,
```



```
    " expo.js " ,  
    " expo.jsx " ,  
    " ts " ,  
    " tsx " ,  
    " js " ,  
    " jsx " ,  
    " json " ,  
    " wasm " ,  
    " svg "  
  ]  
}
```

Utilizando imagens vetoriais

Com as bibliotecas instaladas e a configuração feita, temos suporte para utilizar imagens no formato *SVG*.

No projeto temos uma imagem vetorial na pasta */assets/svg*. Vamos importar essa imagem como um componente *react* e poderemos utilizá-lo em nosso código.

Veja um exemplo abaixo:

```
import ImdLogo from './assets/svg/imd-logo.svg'
```

Daí em diante, é possível utilizar essa imagem como um componente. Porém, é importante informar as dimensões como propriedades do componente.

Para praticar, acompanhe a aula e implemente o código abaixo:

```
import React from 'react';  
import { StyleSheet, Text, View } from 'react-native';  
  
import ImdLogo from './assets/svg/imd-logo.svg'  
  
export default function App() {  
  return (  
    <View style={styles.container}>  
      <ImdLogo width={300} height={300} />  
    </View>  
  );  
}  
  
const styles = StyleSheet.create({  
  container: {
```

```
    flex: 1,  
    backgroundColor: '#fff',  
    alignItems: 'center',  
    justifyContent: 'center',  
  },  
  grande: {  
    fontSize: 50  
  }  
});
```

Resumo

Nesta aula, foi abordado o conteúdo de imagens, exibindo diversas maneiras de trabalhar com elas: padrão 2X/3X, por plataforma, retiradas da internet, *background* e *SVG*. Ademais, vimos como codificar imagens no próprio código.