



# Desenvolvimento para Dispositivos Móveis

## Aula 06 - Componentes Visuais Básicos



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, começaremos a conhecer alguns componentes para criar nossos aplicativos com *react native*.

## Objetivos

- Conhecer como inserir textos na tela
- Conhecer como criar campos que permitam a inserção de dados
- Conhecer como criar botões
- Conhecer como funciona o *switch*

## View/SafeAreaView and Text

Link do video da aula: <https://youtu.be/myR4bRatQhU>

O ponto de partida em um projeto *React Native* é o arquivo "*App.js*", e é ele que iremos percorrer nesta aula.

## Ajustando o arquivo

O arquivo em questão já vem com um conteúdo padrão. Porém, para melhor entendimento, iremos limpar o arquivo deixando-o como se vê abaixo:

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

export default function App() {
  return (
    <Text>Olá!</Text>
  );
}
```

## Conhecendo alguns componentes

O único componente deixado foi o '**Text**', que serve para inserir um texto na tela, similar à tag '*p*' do *HTML*. Porém, na tela o texto sumiu e, para resolver essa questão será criado um componente.

Acompanhando a aula, implemente o componente abaixo:

```
import { StatusBar } from 'expo-status-bar';
```

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

export default function App() {
  return (
    <SafeAreaView>
      <Welcome />
    </SafeAreaView>
  );
}

function Welcome() {
  return (
    <View>
      <Text>Bem vindo</Text>
    </View>
  )
}
```

O componente '**View**' encapsula outros componentes, tornando-se similar à tag '*div*' no *HTML*. Já o '**SafeAreaView**' é um tipo de '*View*' que encapsula os elementos em uma área segura, para não haver conflito com outros elementos da tela que não fazem parte do aplicativo.

---

### **Nota:**

O '*SafeAreaView*' é uma funcionalidade disponível apenas para plataformas *iOS*. Uma alternativa para o *Android* pode ser implementada conforme o link: [How to use SafeAreaView for Android devices](https://reactnative.dev/docs/safeareaview#android).

---

## **TextInput**

Link do video da aula: <https://youtu.be/Xmcu0ZnY9cE>

---

Nesta aula veremos como criar campos que permitam a inserção de dados.

### **Criando campo**

No react native o campo de inserção de texto é denominado de '**TextInput**', que

funciona de forma similar ao '*Input*' do *HTML*. Acompanhando a aula, implemente, no arquivo "*App.js*", o componente abaixo:

```
function Input() {  
  return (  
    <View>  
      <TextInput>  
  
      </TextInput>  
    </View>  
  )  
}
```

Com esse componente básico inserido no componente principal, o *App*, será possível inserir informações no campo de texto.

## Gerenciando o estado

Embora seja possível ver o texto sendo digitado no campo, ainda não estamos gerenciando o estado. Para fazer isso, utilizaremos um *Hook* denominado de *useState()*, assim como no *React* tradicional.

Para visualizarmos melhor, de maneira provisória, podemos fazer uma estilização da caixa de texto.

Acompanhando a aula, implemente as modificações e ao final seu arquivo deve ficar como se observa abaixo:

```
import React, { useState } from 'react';  
import { SafeAreaView, Text, TextInput, View } from 'react-native';  
  
export default function App() {  
  return (  
    <SafeAreaView>  
      <Welcome />  
      <Input />  
    </SafeAreaView>  
  );  
}  
  
function Input() {  
  
  const [text, setText] = useState('')
```

```
return (  
  <View>  
    <TextInput style={{ height: 40 }} placeholder="Digite aqui seu  
nome" value={text} onChangeText={(text) => setText(text)}>  
    </TextInput>  
    <Text>{text}</Text>  
  </View>  
)  
}  
  
function Welcome() {  
  return (  
    <View>  
      <Text>Bem vindo!</Text>  
    </View>  
  )  
}
```

## Button e Switch

Link do video da aula: [https://youtu.be/7d0XxfQ8i\\_0](https://youtu.be/7d0XxfQ8i_0)

Vamos agora estudar os componentes *Button* e *Switch*.

### Adicionando botão

Para adicionar um botão à tela, criaremos outro componente baseado em uma função que irá retornar um componente **'Button'**, similar à tag *'button'* do *HTML*. Nesse caso também utilizaremos o conceito de *props* para o texto do botão.

Em um botão é importante também monitorar o evento de clique e, para fazer isso, podemos utilizar o evento *'onPress'* e nele posso utilizar códigos *javascript* para comunicar o clique ao usuário.

Veja abaixo um exemplo:

```
<Button onPress={onPressButton} title={props.texto}></Button>
```

### Adicionando o switch

O *react native* também nos fornece um componente que dará uma aparência de marcado ou não marcado, denominado **Switch**.

Para o funcionamento correto, temos de criar um estado que reflita o comportamento desse componente. Nesse caso, utilizaremos novamente o *Hook* `useState()`, que irá passar o valor inicial e controlar o comportamento com o evento `onValueChange`.

Acompanhando a aula e realizando as implementações, seu código deve ficar como se vê abaixo:

```
import React, { useState } from 'react';
import { Button, SafeAreaView, Switch, Text, TextInput, View } from
'react-native';

export default function App() {
  return (
    <SafeAreaView>
      <Welcome />
      <Input />
      <Botoes texto="Clique aqui" />
    </SafeAreaView>
  );
}

function onPressButton() {
  alert('Você clicou!');
}

function Botoes(props) {

  const [ligado, setLigado] = useState(false)

  return (
    <View>
      <Button onPress={onPressButton} title={props.texto}></Button>
      <Switch onChange={() => setLigado(!ligado)}
value={ligado}></Switch>
    </View>
  )
}

function Input() {

  const [text, setText] = useState('')

  return (
```

```
    <View>
      <TextInput style={{ height: 40 }} placeholder="Digite aqui seu
nome" value={text} onChangeText={(text) => setText(text)}>
    </TextInput>
    <Text>{text}</Text>
  </View>
)
}

function Welcome() {
  return (
    <View>
      <Text>Bem vindo!</Text>
    </View>
  )
}
```

## Resumo

Nesta aula, vimos como trabalhar com alguns componentes do *react native*, sejam eles para inserir texto na tela, seja para criar caixas de inserção de texto ou até mesmo botões e componentes de seleção. Ademais, vimos como gerenciar estados e utilizar as *props*.