



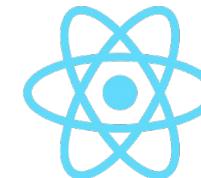
CCP150

DES. DE APLICATIVOS MÓVEIS

Prof. ISAAC
isaacjesus@fei.edu.br

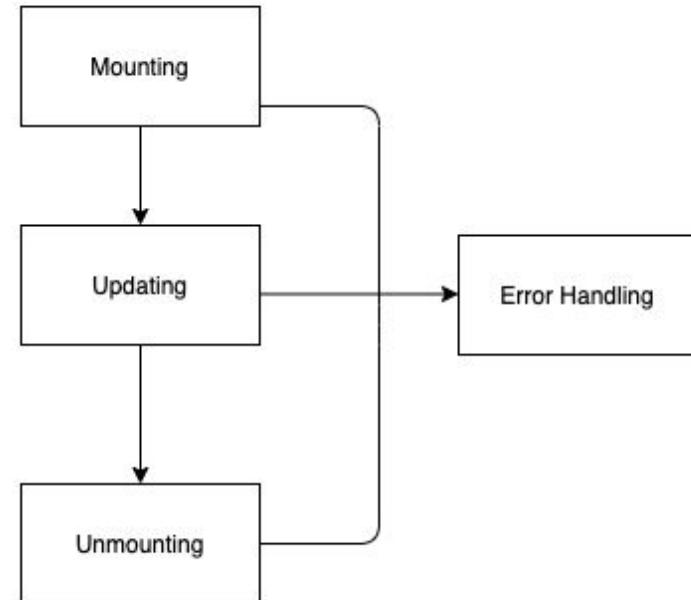
DESENVOLVIMENTO DE APLICAÇÕES MOBILE

Ciclo de Vida de Componentes

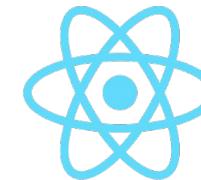


React Native

- Um componente passa pelas seguintes fases:
 - Montagem (*mounting*)
 - Atualização (*updating*)
 - Desmontagem (*unmounting*)
 - Tratamento de Erro (*error handling*)



Ciclo de Vida de Componentes

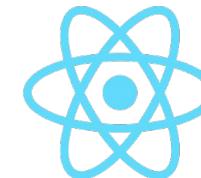


React Native

- **Montagem:** Componente está pronto para ser montado no DOM do navegador
 - Métodos:
 - **constructor:** primeiro método executado
 - **render:** monta o componente no navegador
 - **componentDidMount:** é executado depois que o componente foi montado no DOM

<https://snack.expo.dev/@isaacjesus/3d8707>

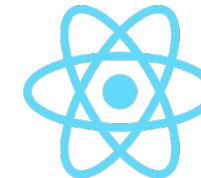
Ciclo de Vida de Componentes



React Native

- **Atualização:** O componente pode ser atualizado pelo envio de novas ***props*** ou pela atualização do seu ***state***
 - Métodos:
 - ***shouldComponentUpdate***: Diz se o componente deve ser renderizado novamente ou se essa ação deve ser ignorada
 - ***componentDidUpdate***: É executado depois que o componente é atualizado no DOM

Ciclo de Vida de Componentes

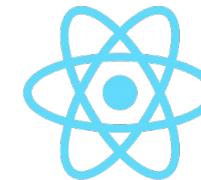


React Native

- **Desmontagem:** O componente não é mais necessário e será desmontado do DOM
 - Métodos:
 - ***componentWillUnmount***: último método no ciclo de vida. Executado imediatamente antes de o componente ser removido do DOM

<https://snack.expo.dev/@isaacjesus/aula06---desmontagem>

Ciclo de Vida de Componentes

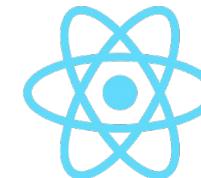


React Native

- **Erro:** Tratamento de erros
 - Métodos:
 - ***componentDidCatch***: usado para localizar erros e tratá-los.
Ajuda a usar qualquer bloco try / catch para lidar com erros

Inserindo imagens

Imagens



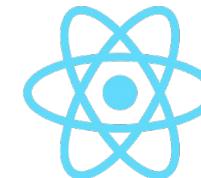
React Native

- Podemos incluir imagens em nossas aplicações!
- Componentes:

```
<Image source = {require('caminho_imagem')}>
```

```
<ImageBackground source = {require('caminho_imagem')}>
```

Imagenes



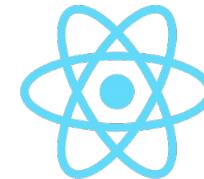
React Native

- Props:
 - **source:**

<https://reactnative.dev/docs/image/#source>

Botões customizados

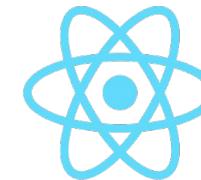
Botões customizados



React Native

- No React Native existe o componente **Button**, mas na prática quase sempre usamos **TouchableOpacity**, **TouchableHighlight** ou **Pressable** porque são botões que podemos customizar.
- **Button** tem estilo limitado: o design varia entre Android e iOS, e você não consegue padronizar bem.

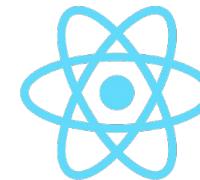
Botões customizados



React Native

- **TouchableOpacity**, **TouchableHighlight** e **Pressable** são mais flexíveis: você pode colocar qualquer componente dentro (texto, ícone, imagem).
- Permitem personalizar completamente o estilo (cores, bordas, sombras, espaçamento).
- Dão acesso a efeitos visuais (opacidade, highlight, mudanças dinâmicas com pressed).

Botões customizados

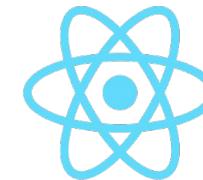


- **TouchableOpacity**: o botão fica mais transparente (reduz opacidade) quando pressionado.

<https://snack.expo.dev/@isaacjesus/847ca3>

<https://reactnative.dev/docs/touchableopacity>

Botões customizados



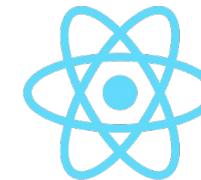
React Native

- **TouchableHighlight**: o botão fica com um fundo colorido (highlight) enquanto está pressionado.

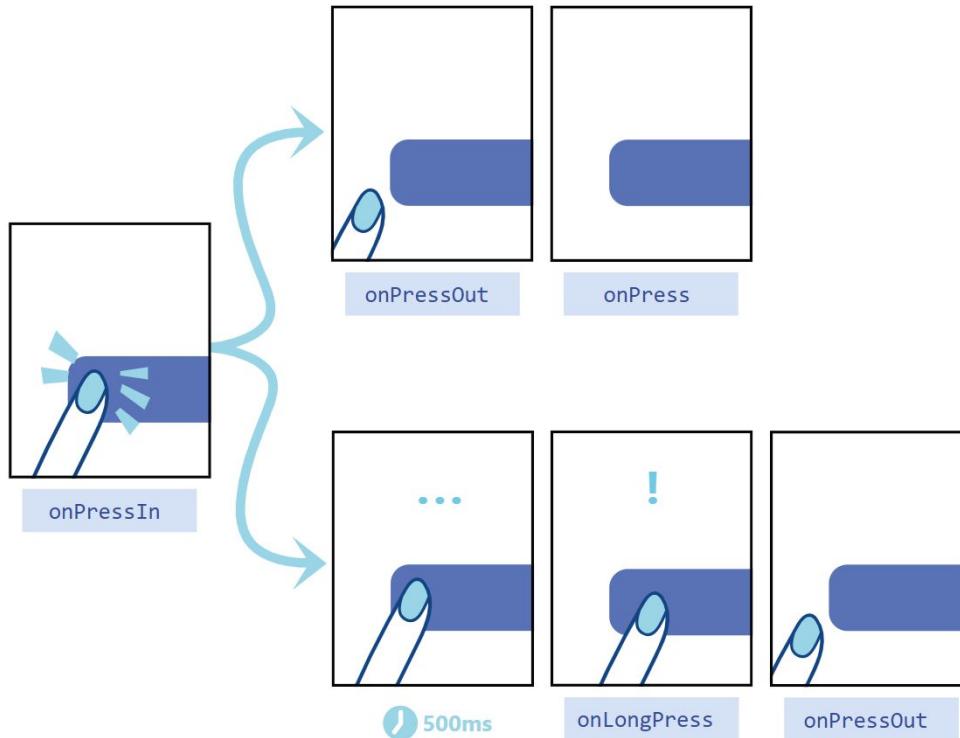
<https://snack.expo.dev/@isaacjesus/aula06---touchablehighlight>

<https://reactnative.dev/docs/touchablehighlight>

Botões customizados



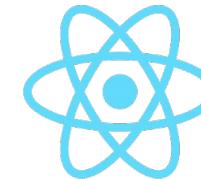
React Native



- **Pressable:** mais flexível, permite detectar diferentes estados: pressionado, focado, clicado, etc.

<https://reactnative.dev/docs/pressable>

Botões customizados



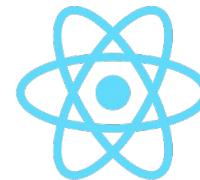
React Native

- **Pressable:**
- Entre no site do React Native, pegue o exemplo do Pressable e coloque no Snack para testar.

<https://reactnative.dev/docs/pressable>

Outros componentes de entrada

Switch



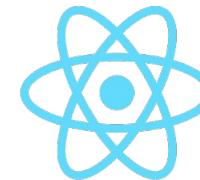
React Native

- O Switch é um componente usado para representar uma chave de ligar/desligar.

<https://snack.expo.dev/@isaacjesus/aula06---switch>

<https://reactnative.dev/docs/switch>

Picker

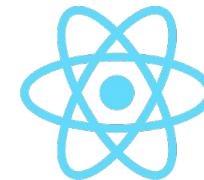


React Native

- O Picker é usado para criar um menu suspenso (dropdown) que permite ao usuário selecionar uma opção.

<https://snack.expo.dev/@isaacjesus/aula06---picker>

Slider



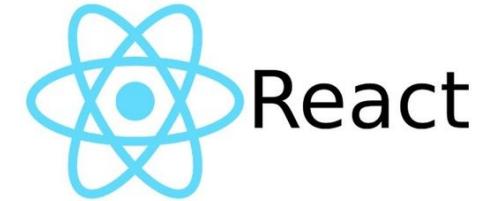
React Native

- O Slider permite ao usuário selecionar um valor dentro de um intervalo deslizando o botão.

<https://snack.expo.dev/@isaacjesus/aula06---slider>

Exercícios

Exercício 01

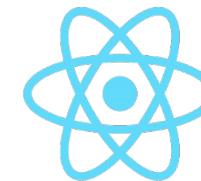


Crie uma aplicação React Native de um **cronômetro**. O objetivo é que o usuário possa iniciar, pausar e zerar a contagem do tempo, com a exibição em tempo real.

Requisitos:

- Controle de Tempo: O cronômetro deve ter 3 botões: Iniciar, Pausar e Zerar a contagem.
- Formato de Exibição: O tempo deve ser exibido em um formato legível, como **HH:MM:SS** (horas, minutos, segundos).
- Inserir imagem e botões customizados.

Exercício 02



React Native

Construir um App de relógio digital (Hora e Data)

- Inserir imagem e botões customizados.