

CCP150

DES. DE APLICATIVOS MÓVEIS

Prof. Isaac
isaacjesus@fei.edu.br



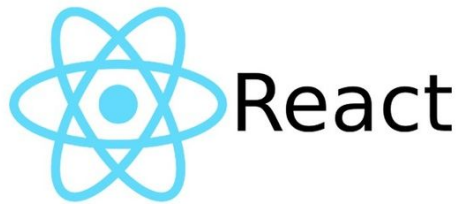
DESENVOLVIMENTO DE APLICAÇÕES MOBILE



MODELOS DE DADOS:

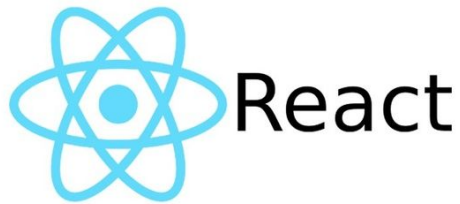
props

Componentes e *props*



- **Componentes aceitam entradas arbitrárias:**
 - **Funcionam como argumentos em uma Função JavaScript**
 - Chamadas ***props*** (*propriedades*)
 - Os ***props*** são **somente leitura!**
 - Os ***props*** não devem ser modificados pelos Componentes (funções puras)

Componentes e *props*



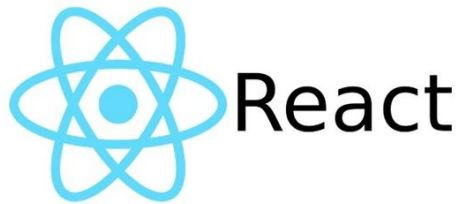
- Exemplo com **Componente de Função:**

<https://codepen.io/danilo-perico/pen/XWKWqoB?editors=1010>

<https://codepen.io/Isaac-Jesus-Silva/pen/KKYZrQq?editors=1010>



Componentes e *props*



- Exemplo com **Componente de Classe:**

<https://codepen.io/danilo-perico/pen/ZEOEoNo?editors=1010>

<https://codepen.io/Isaac-Jesus-Silva/pen/WNWdYJp>

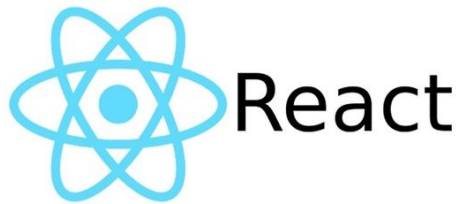


Componentes e *props*

- O uso de *props* permite que você crie um único componente que é usado em muitos lugares diferentes em seu aplicativo, com propriedades ligeiramente diferentes em cada lugar



Componentes e *props*



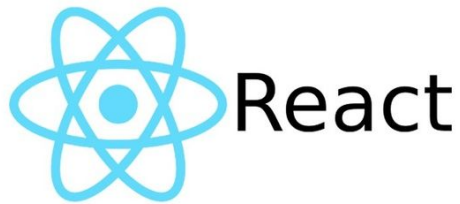
- Exemplos - Composto componentes:

<https://codepen.io/danilo-perico/pen/JjKjadL?editors=1010>

<https://codepen.io/danilo-perico/pen/JjKjZpo?editors=1010>



Exercício



Crie um componente **Card** que receba as seguintes props:

- nome
- idade
- profissão

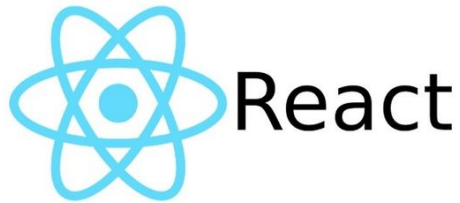
Crie um componente **App** que renderize pelo menos 3 Cards com conteúdos diferentes para os props do **Card**.





MODELOS DE DADOS: *Estados*

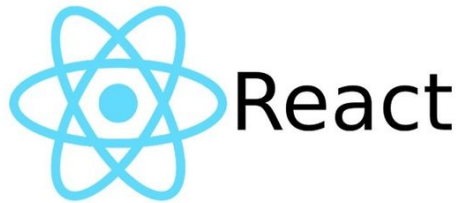
Estados



- São como os **atributos** de uma classe em Java ou C++
- A diferença é que quando **o valor do estado é alterado**, o componente é **renderizado novamente automaticamente**
- Representam o estado de um componente

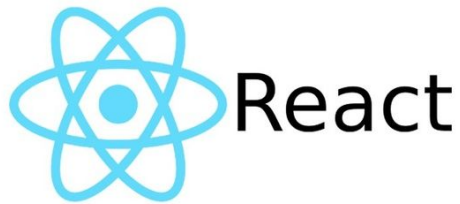


Estados



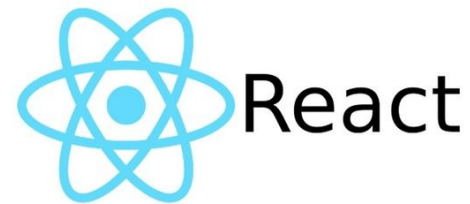
- São como os **atributos** de uma classe em Java ou C++
- A diferença é que quando **o valor do estado é alterado**, o componente é **renderizado novamente automaticamente**
- Representam o estado de um componente

Estados



- **this.state** é o atributo que guarda todos os estados do componente
- **this.setState** é um método do componente que faz a atualização dos estados

Estados



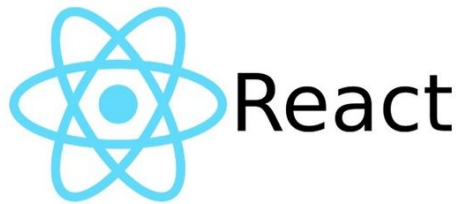
Exemplo:

this.state:

<https://codepen.io/danilo-perico/pen/JjKoGee?editors=1010>



Estados



Exemplo: **this.setState:**

<https://codepen.io/danilo-perico/pen/JjvaLQQ>

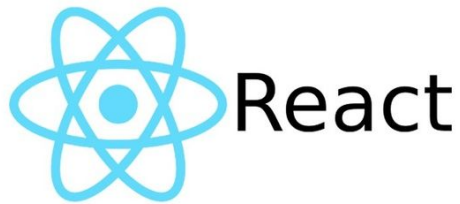
<https://codepen.io/Isaac-Jesus-Silva/pen/NWmyrWP>





Eventos

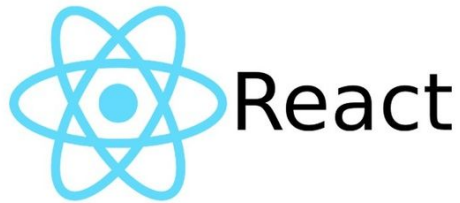
Eventos



- São **mecanismos utilizados para invocar funções**
- Muitas vezes associados **às interações do usuário**
- Utilizam o conceito de ***callbacks***



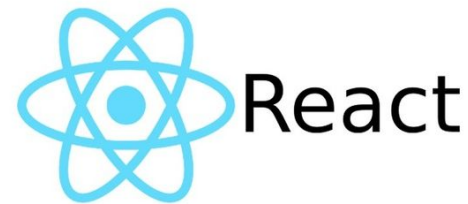
Callbacks



- **Funções que são passadas para outras funções**
- São utilizados pelas chamadas **High Order Functions**
- Normalmente são passados como parâmetros para outras funções
- As **High Order Functions** são responsáveis por invocar os **Callbacks**



Callbacks

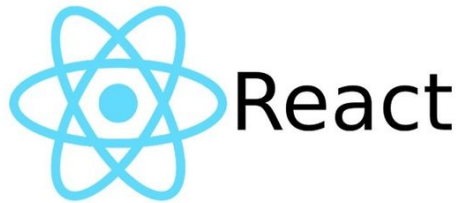


<https://codepen.io/Isaac-Jesus-Silva/pen/xxeYOrz>

<https://codepen.io/Isaac-Jesus-Silva/pen/PogQzpY>



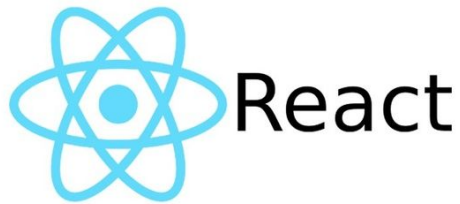
Eventos no React



- Manipular eventos em elementos React é muito semelhante a manipular eventos em elementos do DOM. Existem algumas diferenças de sintaxe:
 - Eventos em React são nomeados usando **camelCase** ao invés de letras minúsculas
 - Exemplo: **onClick** ao invés de **onclick**



Eventos no React



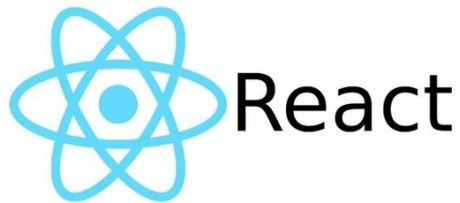
- Exemplos:

<https://codepen.io/Isaac-Jesus-Silva/pen/oNOELyV>

<https://codepen.io/Isaac-Jesus-Silva/pen/eYoVzQM>



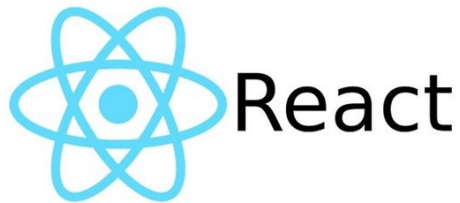
Exemplo



Crie uma aplicação React Web e no componente principal App faça:

- Renderize dois botões. O primeiro mostrando o rótulo "Oi" e o segundo mostrando o rótulo "Tchau".
- Vincule ao primeiro botão um método que faz o alerta do texto: "Oi"
- Vincule ao segundo botão um método que faz o alerta do texto: "Tchau"

Eventos no React



- Exemplos:

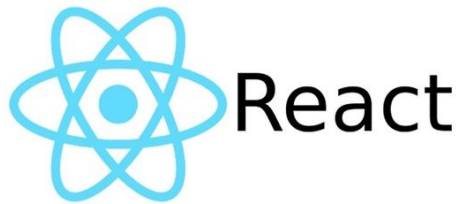
onChange

<https://codepen.io/Isaac-Jesus-Silva/pen/poBKzeZ>

<https://codepen.io/Isaac-Jesus-Silva/pen/VwNXMdz>



Exercício



Crie uma aplicação React Web com Programação Orientado a Objetos:

- Receba dois valores numéricos.
- Crie 4 botões para soma, subtração, multiplicação e divisão.
- Realize o cálculo apresentando a resposta referente a operação escolhida, exemplo:
 - A soma de $23 + 57 = 80$

