

React Context

O que vamos ver hoje?

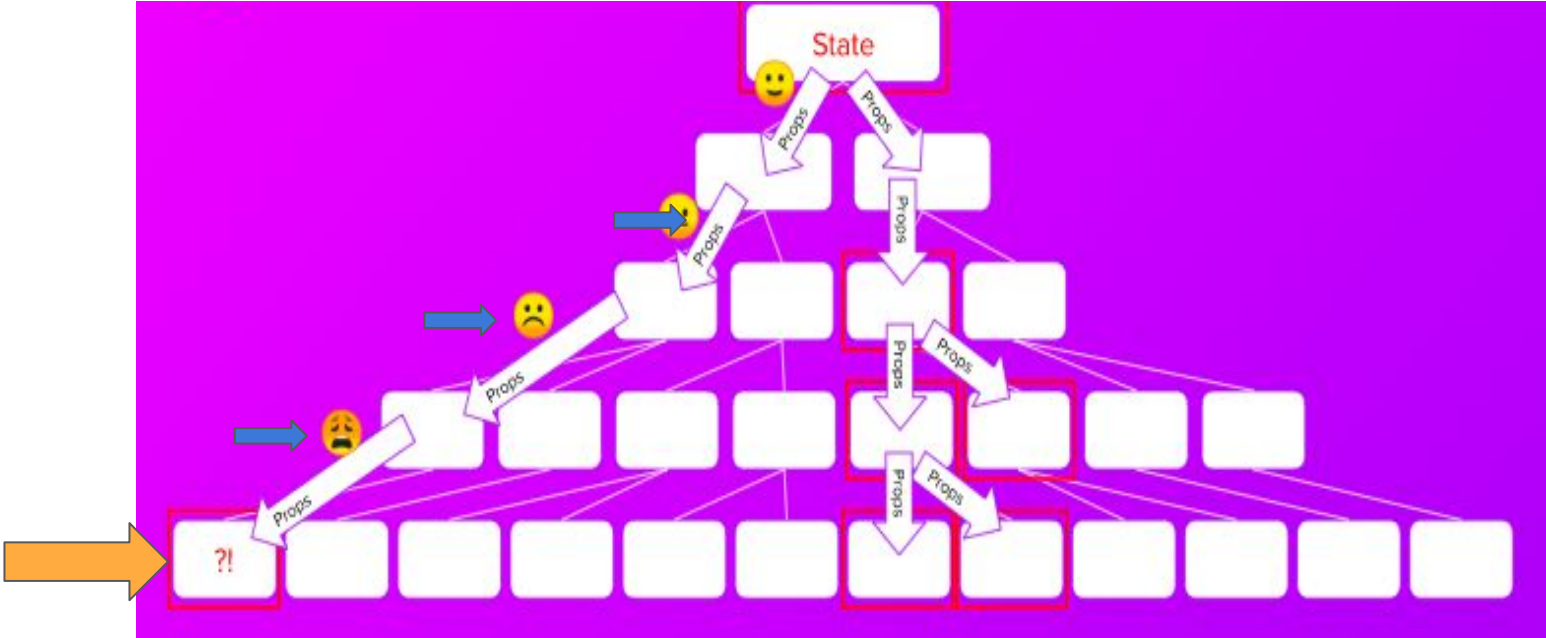
- O que é Context?
- Criando um Contexto
- Quando usar Context?
- Exemplo Prático

O que é o Context?

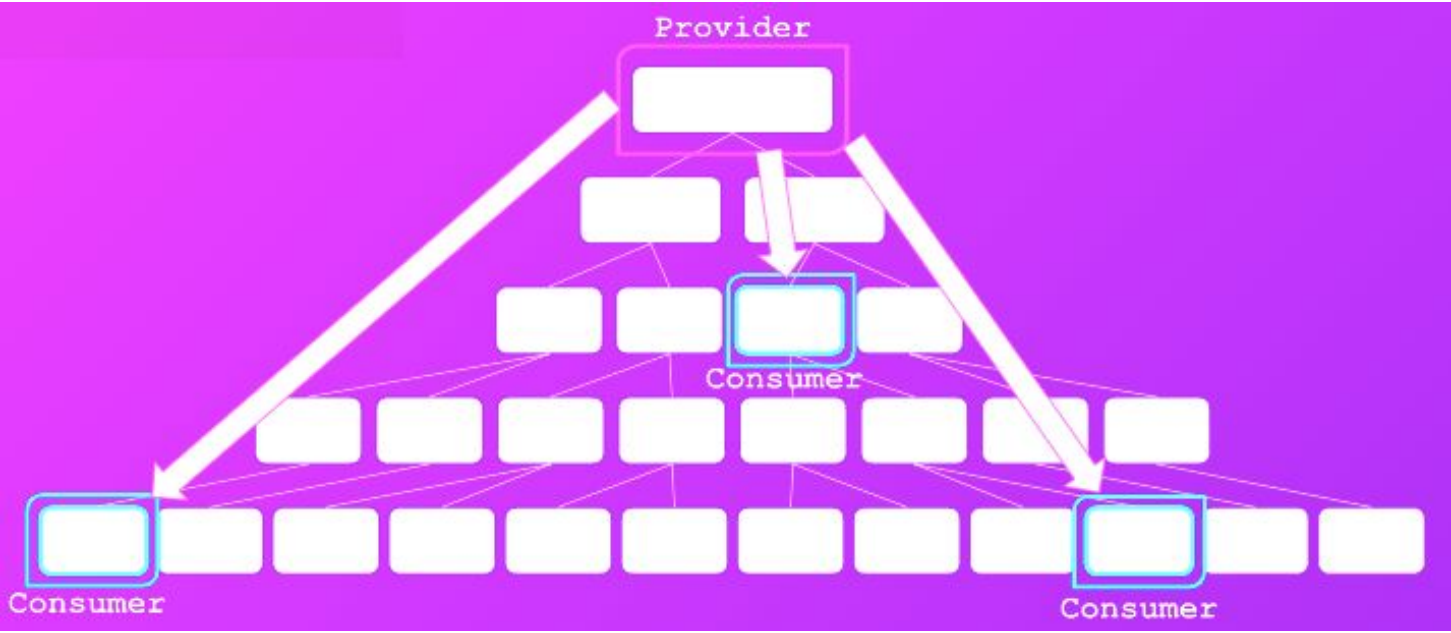
React Context - Definição

- O React Context nos permite criar um **contexto**, que é uma forma de passar dados para componentes **sem precisar usar props**
- Todo contexto do React tem 2 peças importantes:
 - **Provider:** recebe dados e disponibiliza para os Consumers
 - **Consumer:** tem acesso aos dados de um provider

React Context - Definição



React Context - Definição



React Context - Definição

- Criamos um **Provider** que recebe um dado e está **acima** dos componentes que precisam acessá-lo
- Ao invés de passar o dado de componente em componente usamos um **Consumer** que nos permite acessar o dado **diretamente**



Criando um Contexto

Criando um Contexto

- O **context** é o local onde os dados serão guardados
- É comum que ele fique em outro arquivo
- A sua criação se dá da seguinte forma:

```
1 export const ContextoBola = React.createContext()
```

Criando o Provider

- O **provider** é o provedor dos nossos dados, ele é criado **usando o contexto**
- Precisa receber uma prop **value** que representa as informações contidas no contexto
- Esse valor pode ser de **qualquer tipo**: string, number, array, objeto, função, etc.

Criando o Provider


- Ele é um componente que disponibiliza o valor para todos os seus filhos

```
1 import { ContextoBola } from "../context"
2
3 export default function App() {
4   return (
5     <ContextoBola.Provider>
6       { /*Todos os componentes aqui dentro*/ }
7       { /*terão acesso ao contexto*/ }
8     </ContextoBola.Provider>
9   )
10 }
```

Colocando dados no Provider

- Precisamos passar **um valor** para o nosso Provider disponibilizar
- Vamos também adicionar **um novo componente** como filho do nosso provider

```
1 import PaiSemBola from "../components/PaiSemBola"
2 import { ContextoBola } from "../context"
3
4 export default function App() {
5   const bola = {
6     tipo: "basquete",
7     imagem: "https://www.bolas.com/basquete.png"
8   }
9
10  return (
11    <ContextoBola.Provider value={bola}>
12      <PaiSemBola />
13    </ContextoBola.Provider>
14  )
15 }
```



Componente intermediário



- O componente **PaiSemBola** não tem lógica, só serve para adicionar estilos
- Agora vamos ver o componente **FilhoComBola**, é dentro dele que iremos acessar os dados do contexto

```
1 const PaiSemBola = () => {  
2   return (  
3     <WrapperBola>  
4     → <FilhoComBola />  
5     </WrapperBola>  
6   )  
7 }
```

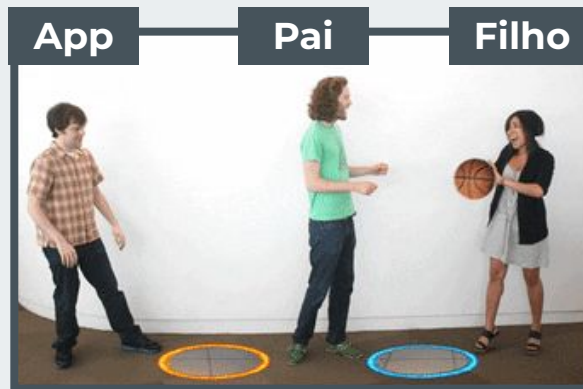
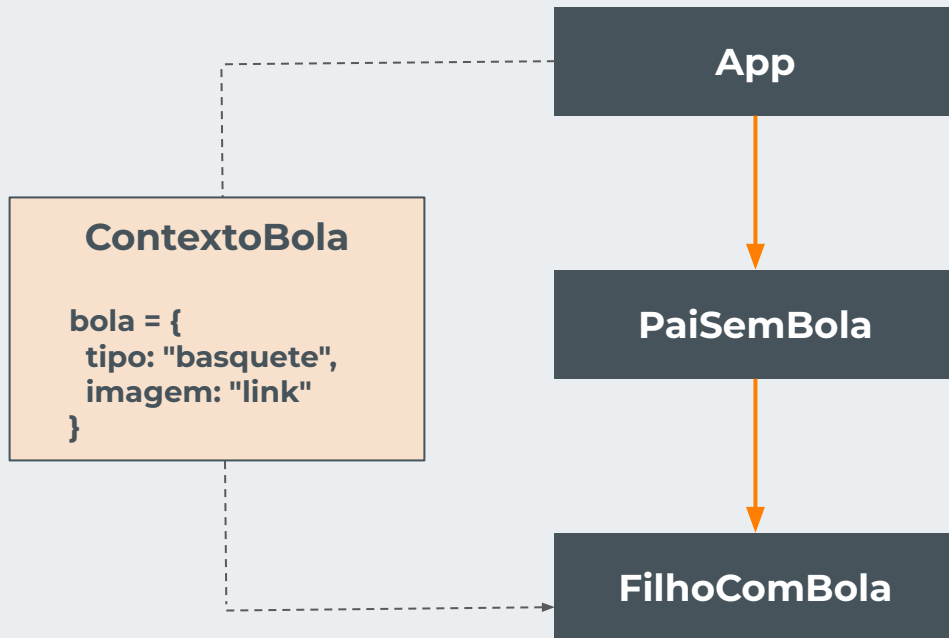
Acessar o Contexto no Consumer

- Agora que estamos no componente que precisa de fato acessar os dados vamos ver como podemos ter acesso ao contexto:

- Ao combinar o **contexto** e o **hook** obtemos o **valor armazenado no contexto**

```
1 import React, {useContext} from "react"
2 import {ContextoBola} from "../context"
3
4 const FilhoComBola = () => {
5   const bola = useContext(ContextoBola)
6   return (
7     <div>
8       <h1>`Bola: ${bola.tipo}`</h1>
9       <img alt="bola" src={bola.imagem} />
10     </div>
11   )
12 }
13 }
```

Exemplo da Bola



Demonstração 

Quando usar Context?



Context - Ressalvas

- O Context é uma ferramenta robusta para problemas robustos, **nem sempre ela é necessária**
- Pode ser um **exagero** no caso de projetos mais simples, demanda um trabalho de setup e aumenta a complexidade do seu projeto
- Quanto maior a aplicação, mais ela provavelmente se beneficiaria do context

Context - Ressalvas

- Se você está passando props em até **2 níveis**, não precisa de context!
- Atenção para a **abrangência do Provider**, ele não necessariamente precisa envolver a aplicação toda
- Evite usar context em lugares onde o **estado local é o suficiente**, existem informações que são necessárias em um componente apenas

Usos comuns do Context

Usos Comuns do Context

- Linguagem do site
- Temas dos componentes
- Material `<ThemeProvider>`
- Router `<BrowserRouter>`