



React para iniciantes





<title>

AGENDA DO </> </> </> MÓDULO

</title>



Agenda do módulo



Anotações

{Introdução a contextos}

{Construindo contexto de lista de favoritos}

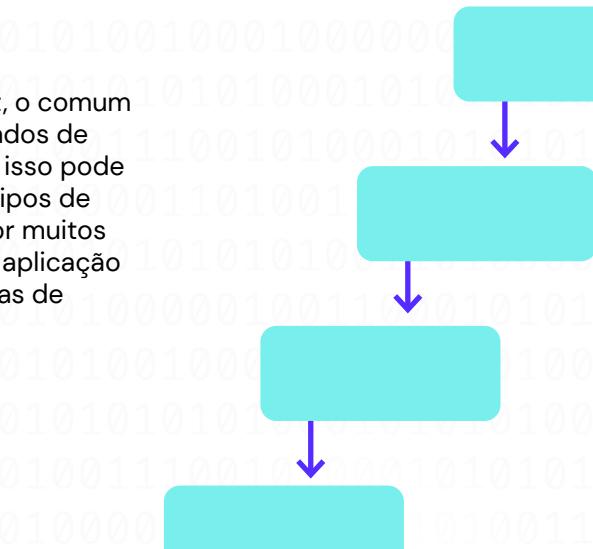
{Conectando a aplicação com o contexto}

{Finalização da aplicação}

{Deploy da aplicação}



Em uma aplicação em React, o comum é que os dados sejam passados de pai para filho via props, mas isso pode não ser tão fácil em certos tipos de props, como as utilizadas por muitos componentes dentro dessa aplicação (*prop drilling*). As preferências de locais são um exemplo.



<title>

MISSÃO DO MÓDULO

</title>

Construir e conectar a aplicação Filmz usando **contextos** e finalizar o projeto.



{01.}

Introdução a contextos

O contexto em React fornece uma maneira de passar dados pela árvore de componentes sem ter a necessidade de passar props manualmente em todos os níveis da árvore.

Dentro de uma aplicação em React, os dados são passados de “cima para baixo”, ou seja, “de pai para filho” por meio de props. Mas essas condições dependem do tipo de prop, como, por exemplo:

- › preferência de localidade;
- › tema de interface do usuário, entre outros — assim são exigidos muitos componentes dentro de um aplicativo.

Em um contexto, podem ser compartilhados diversos elementos:

- › funções
- › constantes
- › strings
- › arrays
- › estados etc.



<title>

PULO DO GATO

</title>

- Hooks são outra forma de chamar os contextos.

Usamos Contexto para compartilhar uma informação global entre diferentes componentes, independentemente de hierarquia, e em diferentes lugares de um aplicativo.

Um outro bom exemplo é o carrinho de compras dos e-commerces.

Em e-commerces, você pode acessar o carrinho de qualquer lugar, e geralmente adicionar itens nele, originando diversas páginas diferentes. Neste caso, um contexto seria bem útil.





Para ter sentido,

precisa ter

contexto.

contexto

contexto

contexto

contexto

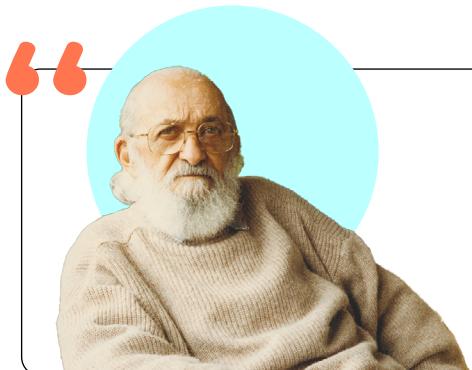
contexto

contexto

contexto

React para iniciantes: Contextos no React





+

+

+

+

+

+

A teoria sem a prática vira
‘verbalismo’, assim como a prática
sem teoria vira ativismo.
No entanto, quando se une a prática
com a teoria, tem-se a práxis,
a ação criadora e modificadora
da realidade.

Paulo Freire

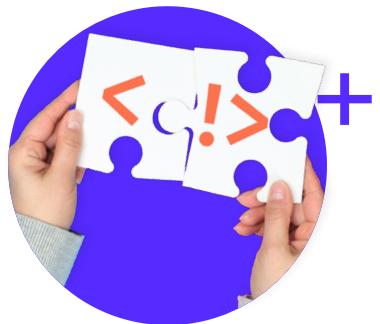


<title>

MÃO + + + NA MASSA

</title>

Crie o arquivo para o seu contexto
e também a interface que ele deve
implementar.





{02.}

Construindo contexto de lista de favoritos

Para adicionar funcionalidades ao contexto, vamos primeiro criar uma função, ela é quem irá fornecer as funcionalidades para o contexto. O nome dessa função é Provider. E seu papel será prover as funcionalidades do contexto.

Para dar um nome semântico, ela será chamada de “[WishlistProvider](#)”. Pois será a função provedora do contexto de WishList.



<title>

PULO DO GATO

</title>

- O que é criado dentro do contexto, para **ser usado em qualquer lugar da aplicação**, precisa estar **dentro da propriedade values** do componente Provider daquele contexto.
- Não armazene informações sensíveis no **localStorage**, pois qualquer pessoa pode ter acesso a elas.

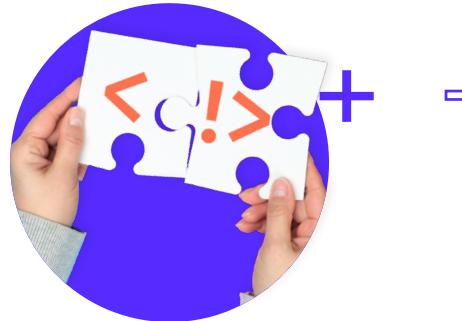




<title>

MÃO + + + NA MASSA

</title>



- › Crie o **Provider** para o seu contexto, assim como sua interface.
- › Depois, implemente-o e, por fim, crie as funções do seu contexto.

{03.}

Conectando a aplicação com o contexto



Para utilizarmos o contexto, devemos pegar o retorno da função useContext, passando o nosso contexto de Wishlist como parâmetro e desestruturar o retorno dela.

O código vai ficar dessa forma:

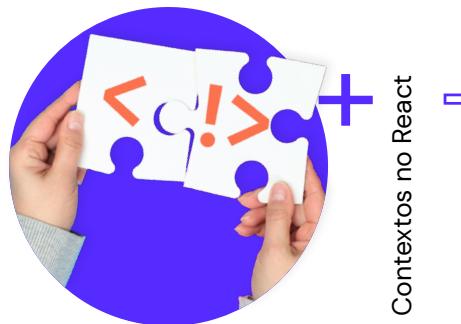
```
const { wishlist } = useContext(WishlistContext)
```

Existe um padrão nas funções do React.
Sempre começam com "use", e depois o nome dela.

```
<title>
```

MÃO + + +
NA MASSA

```
</title>
```



Refine o seu contexto e implemente-o na página Home.



{04.}

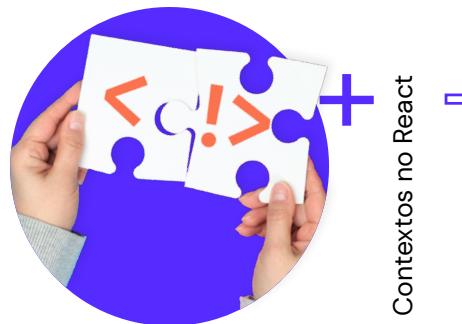
Finalização da aplicação

<title>

MÃO + + NA MASSA

</title>

- › Agora chegou a sua vez de implementar o seu contexto criado nas demais páginas da sua aplicação.
- › Seguindo os passos descritos na aula, faça o mesmo no seu arquivo pessoal do projeto.





{05.}

Deploy da aplicação

Para publicar o seu site, utilize a [Vercel](#).

Vercel é a empresa que criou o Next (derivação do React), que manda muito bem quando se fala em deploy.

Você pode publicar suas aplicações de forma fácil e gratuita.



<title>

PULO DO GATO

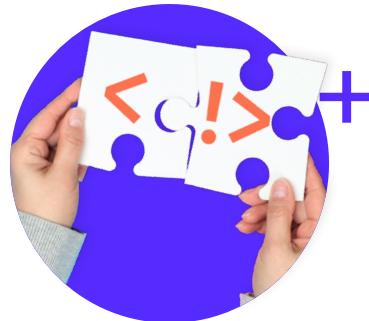
</title>

- Para facilitar sua jornada, crie sua conta no site da Vercel, usando o seu GitHub.



<title>

MÃO + + + NA MASSA



</title>

Faça o deploy da sua aplicação e incremente o seu portfólio.

<title>

DESAFIO CONQUER

</title>



```
if (challenge)  
  // start
```

<body>

- › Faça uma melhoria para o componente Header, para que o ícone de **Minha lista** mostre ao lado quantos filmes temos adicionados na WishList.



</body>



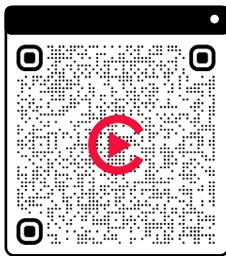
<title>

QUERO MAIS

</title>



Leia o artigo da documentação de React e descubra mais sobre Contextos



Aprenda a deixar o site “Filmz” responsivo



```
if (interested)  
// gimmemore
```



Anotações

-
-
-
-
-
-

