



Boas-vindas! Caro aspirante a dev!

Aula 08 - Conceitos Avançados - Parte 2:

Esta é nossa oitava aula do módulo de React, nela abordaremos os seguintes tópicos:

- useMemo
- useRef
- useImperativeHandle
- useId
- useTransition
- useDeferredValue
- useCustom
- *Exercícios teóricos e práticos para entrega*

Ao final desta aula, você estará apto(a), a:

- Evitar renderizações desnecessárias em elementos filhos com a memorização;
- Realizar operações imperativas em componentes react;
- Utilizar a simultaneidade da UI do react 18;
- Criar seus próprios hooks personalizados;
- Realizar os exercícios teóricos e/ou práticos propostos.

Exercícios:

1. Para que serve o recurso de memorização do react?
2. Quando se deve utilizar o recurso de simultaneidade do react 18?
3. Realizar a atividade prática solicitada:
 - a. Crie um hook customizado, que recebe um array de produtos, cujos objetos contêm o nome do produto, preço unitário e quantidade; e um array de parcelas que possuem objetos com o valor da parcela. Seu hook personalizado deve retornar o valor total dos itens (da venda), o total em parcelas, e um valor booleano que indica que o valor total de parcelas é igual ao total da venda, sempre que o array de itens ou o array de parcelas mudar, este hook deve reagir e retornar novos valores.

Em síntese:

Ao final desta aula você deverá ter compreendido a maioria dos hooks nativos do react, que não são muito conhecidos mas que podem trazer melhorias significativas na performance de sua aplicação. Será possível reduzir drasticamente o número de renderizações desnecessárias com o recurso de memorização de componentes filhos, e utilizando o recurso de adiamento de renderizações pesadas adicionado no react 18. Além disso, você aprenderá a criar seus próprios hooks, fazendo com que muitas lógicas de estados sejam compartilhadas facilmente entre diferentes componentes.



Referências:

- [API dos Hooks](#)
- [useTransition e useDeferredValue](#)
- [useId](#)
- [useMemo](#)
- [useImperativeHandle](#)