

🧠 Fundamentos do React

- O que é o React e para que ele é usado?
- R: É uma biblioteca de componentes e funções reativa javascript criado pelo grupo Meta.
- Qual a diferença entre uma função JavaScript comum e um componente React?
- R: É o lance da reatividade. O Javascript não re-renderiza a tela novamente quando uma informação é alterado, já o React ele atualiza a tela quando uma informação foi alterada de um useState e react precisa fazer a reconciliação entre o VIRTUAL DOM e o REAL DOM.
- VIRTUAL DOM: É uma cópia da ÁRVORE de tags HTML que temos no navegador (REAL DOM)
- O que são props e como são usadas?
- R: Props são informações que são passadas de um componente para o outro. Exemplo... temos um componente chamado MENU que precisamos para uma informação que esta nele para outro componente filho chamado BOTÃO. Isso se chama PROPS DRILLING.
- O que é o estado (useState) e por que é importante?
- R: useState é uma variável reativa. Ou seja, toda vez que ela for alterada o react vai re-renderizar o componente inteiro novamente.
- Como funciona o ciclo de vida de um componente (hooks como useEffect)?
- R: O ciclo de vida funciona da seguinte forma: Se o array de dependencia está vazio... o useEffect é disparado quando o componente é montado. Se existe alguma variavel no array de dependencia, ele só é disparado quando aquela variável alterar.

🍷 Estrutura e JSX

- O que é JSX?
- R: É o padrão de linguagem do React único que mistura HTML com JAVASCRIPT no mesmo código. Quando é extensão TSX é porque o arquivo é em TYPESCRIPT.
- Por que não podemos retornar múltiplos elementos sem envolver em um elemento pai?
- R: Cada retorno do react, é necessário apenas UM ITEM do DOM, ou seja apenas um PAI, e daí o resto dos elementos dentro desse PAI.
- Qual a diferença entre class e className no React?
- R: Por causa dos Class Componentes que existam antes de 2019, o React mudou para className não conflitar com a palavra reservada do Class Componentes que é OOP.
- Como condicionar renderizações (ex: &&, ternário)?
- R: São lugares da tela ou componentes que vocês mostram baseado em uma condição de um valor de um useState reativo.

🔗 Hooks

- Para que serve o useEffect? Dê um exemplo de uso.
- R: É uma função que é considerada um efeito colateral, e ela é disparada ou quando o componente é montado ou quando uma variável de dependencia do array sofre alteração.
- O que é um CustomHook
- R: São funções customizadas e a diferença entre ela e funções PURAS do Javascript nada mais é do que o Custom Hook pode armazenar um estado Reativo e as funções puras não.
- O que o hook useRef faz?
- R: É quando você precisa fazer uma referencia a um item no DOM, pois por algum motivo você vai ter que capturar alguma informação do item no DOM (Valores, tamanhos, item dentro dele, posição X Y.)
- Para que serve o useContext?
- R: O useContext é para criar um ESTADO GLOBAL no react e nós conseguirmos armazenar e acessar as informações que estão neste estado.

📦 Gerenciamento de estado

- Qual a diferença entre estado local (useState) e contexto global (zustand)?
- R: O useState é para controle de estado local (No componente) já o context global (Zustand, Redux ou Recoil) é para armazenar estados de forma global.
- Quando faz sentido extrair o estado para um contexto/global?
- R: Quando você precisa passar um estado para um componente que esta acima de dois níveis.

🔗 Comunicação com APIs

- Como fazer uma requisição HTTP em React?
- R: Precisa de uma função do navegador para fazer uma request HTTP. (Fetch ou XMLHttpRequest)
- O que é o axios?
- R: O axios é uma abstração do FETCH.
- O que são chamadas assíncronas?
- R: São chamadas externas elas vão retornar uma PROMISE, ou seja uma promessa de resultado seja ela NULA ou não. Se você usar um AWAIT na frente de uma chamada ASSINCRONA, o JS espera o resultado dessa promise antes de continuar para o resto do código.
- Para que serve a biblioteca React Query ou SWR?
- R: São bibliotecas para armazenar dados em um estado global e apenas o React Query possui funções que podem enviar dados para uma requisição HTTP.

🌿 React Native Específico

- Qual a diferença entre React e React Native?
- R: React é para aplicações WEB e o react native apesar de fazer WEB e DESKTOP ele é voltado para aplicações MOBILE.
- Quais são os principais componentes do React Native?
- R: Text, View, Button, TouchableOpacity, TouchableHighlight, ScrollView, Flatlists, Scrolllist, Modal.
- Como funciona o estilo no React Native? (e a diferença para CSS tradicional)
- R: Ou usamos o Stylesheet do próprio React Native... ou usaremos o NATIVEWIND.
- Como tratar navegação entre telas no React Native?
- R: Ou você vai usar o EXPO ROUTER ou você vai usar o React Nativigation.
- Como acessar recursos nativos como câmera, GPS, etc.?
- R: Dentro do EXPO temos que chamamos de EXPO SDK e ele tem vários módulos para fazer a tratativa dos módulos nativos.

✅ Testes e Boas Práticas

- Você já escreveu testes para componentes React? Usou que biblioteca?
- R: Estudem sobre VITES, REACT TESTING LIBRARY para testes unitários. E para testes E2E vocês podem conhecer o PLAYWRIGHT.
- Como garantir acessibilidade e boas práticas de usabilidade?
- R: Uma boa pratica é colocar os ARIA LABELS em todos os componentes que exigem interatividade para pessoas que tem deficiência de visão possa utilizar a aplicação.
- Qual a importância da separação de responsabilidades em componentes e funções?
- R: Tanto componentes como funções elas devem ter uma única responsabilidade e não podemos fazer várias coisas ao mesmo tempo dentro dos componentes e funções.

⚙️ Extra (Avançado)

- Como funciona o useMemo e useCallback?
- R: São funções para melhorar a performace da aplicação pois ela armazena os dado na memória.
- O que é memoização de componentes (React.memo)?
- R: É armazenar um componente na memória e ele não será renderiza novamente caso o estado dele não mude.
- Como lidar com performance em listas grandes (FlatList, virtualização, etc.)?
- R: Lembrar do conceito de paginação e o conceito de Infinite Scroll. FlatList e VirtualList.
- O que são "controlled" vs "uncontrolled" components?
- R: Os controlados usando o useState para armazenas as informações que foram utilizadas no componente. Já o não controlado, você não consegue pegar os dados usando um onChange por exemplo e armazenar os dados dentro de um state. Para você pegar os dados do componente, somente usando o inputRef.current.value por exemplo.