

Especialização em Front End I - Avaliação Parcial

Total de pontos 64/100 ?

Terá tempo para completar esta avaliação até ao final da aula. No entanto, isto foi concebido para ser concluído em menos de 2 horas.

Para passar, é necessário ter uma pontuação de 70 ou mais de um total de 100.

Uma vez preenchido o questionário, terá de informar o seu professor para que este possa confirmar que a avaliação foi completada corretamente.

E-mail *

matheusrbferreira@gmail.com

Dados do estudante

0 de 0 pontos

Nome e sobrenome *

Matheus Rayol Ferreira

Hooks ao ataque

23 de 32 pontos



Como posso acessar um elemento DOM usando React? *

7/7

- ☐ document.getElementById("id")
- ☒ Usando "ref" do componente.
- ☐ document.querySelector("query")

Como posso acessar o valor de um useRef? *

8/8

- ☐ console.log(ref)
- ☐ console.log(ref.currentValue)
- ☒ console.log(ref.current)
- ☐ Não se pode acessar o valor.



O que acontece quando se usa o Hook useRef como no exemplo? *

8/8



- ☐ Cria uma referência com um valor nulo e retorna indefinido.
- ☐ Retorna null.
- ☒ Cria e devolve uma referência com um valor inicial nulo.
- ☐ Cria e retorna uma referência ao valor de um objeto HTML.

Para que serve useRef? *

0/9

- ☐ Persistir no HTML durante as renderizações.
- ☐ Para acessar o elemento HTML do componente.
- ☐ Nenhuma delas está correta.
- ☒ Todas estão corretas.

Documentação

22 de 32 pontos



Qual é a forma correta de documentar esta função? *

5/5



```
function buscarSapato ({modelo, numero}) {  
  /* código */  
}
```

- ☐ @param informacao { modelo: string, numero: number }
- ☐ @param informacao { modelo: String, numero: Number }
- ☐ @param {modelo: String, numero: Number } informação
- ☒ @param {modelo: string, numero: number } informação

Quando é que os PropTypes fazem as validações? *

2/2

No momento da execução. ▼



Como posso definir que uma propriedade do meu componente em React recebe um dos dois valores possíveis?

*6/6

```
const meteorologia = ({ prognostico }) => (  
  /* Valores possíveis: "ensolarado" ou "nublado" */  
  <p>{prognostico}</p>  
);
```

- ☐ Usando @param {string} é suficiente.
- ☐ Usando @param {"ensolarado" | "nublado"}
- ☐ Usando PropTypes.exact(["ensolarado", "nublado"])
- ☒ Usando PropTypes.oneOf(["ensolarado", "nublado"])

Utilizando o JSDoc, como indicamos que preferimos não utilizar uma função porque existe uma alternativa mais atualizada?

*5/5

- ☒ Usando @deprecated.
- ☐ Usando @override e mencionando qual a função que a anula.
- ☐ Explica-se em @summary que já não é utilizado.
- ☐ Marcação @all e tudo será apagado.

Os PropTypes podem ser utilizados sem o React. *

0/3

Falso



O que nos diz esta documentação? *

0/7

```
/**
 * @param {number} quantidade
 * @param {string[]} produtos
 * @param {string} codigoDesconto
 * @returns {number}
 */
function calcularCusto(quantidade, produtos, codigoDesconto) { /* código */ }
```

- ☐ É uma função que toma três parâmetros: quantidade, que é um inteiro; elementos, que é um conjunto de strings; discountCode, que é uma string; e retorna um inteiro, ou vírgula, que é o custo total.
- ☐ É uma função que retira três elementos de um parâmetro (objeto): quantidade, que é um número inteiro; elementos, que é um conjunto de strings; discountCode, que é uma string; e retorna um número inteiro, ou vírgula, que é o custo total.
- ☒ É uma função que toma três parâmetros: quantidade, que é um número inteiro; elementos, que é um conjunto de strings; discountCode, que é uma string; e devolve um número inteiro que é o custo total.
- ☐ É uma função que toma três parâmetros: quantidade, que é um inteiro; elementos, que é um conjunto de strings; discountCode, que é um id; e retorna um número com uma vírgula, que é o custo total.

Qual dos seguintes elementos não é um prop-type válido? *

4/4

- ☐ PropTypes.bool
- ☐ PropTypes.func
- ☒ PropTypes.enum
- ☐ PropTypes.object

React Query

19 de 36 pontos



Se tenho uma Mutation que modifica o mesmo recurso (neste caso, "users"), *4/4 qual é a forma correta de nossa useQuery para exibir a informação atualizada?



- ☒ Utilizamos "refetchQueries" fornecidas por "useQueryClient" no "onSuccess" da mutation.
- ☐ Utilizamos "setQueryData" em "onMutate" da nossa "useMutation".
- ☐ Utilizamos "setQueryData".
- ☐ Criamos um "useState" com os dados de "useQuery", e atualizamos em "onSuccess" da nossa "useMutation".



O que é que falta neste código? *

0/6



- ☐ Falta o "usuarios[0].id" no array queryKey, para aproveitar o cache.
- ☐ Não falta nada, mas as queries devem ser unificadas (usando UseQueries).
- ☒ Não falta nada.

Como instalamos o React Query? *

2/2

- ☐ npm install react query
- ☐ npm install @react-query
- ☒ npm install react-query
- ☐ npm install @react/query



O que significa se o estatuto de uma query é "stale"? *

6/6

- ☐ Significa que a query é congelada (já não pode ser atualizada).
- ☒ Isto significa que os dados da query não são "frescos" (potencialmente não atualizados).
- ☐ Significa que espera que outra ação seja executada.

Para que utilizamos o React Query? *

0/4

- ☐ Para melhorias de desempenho em banda larga e consumo de memória.
- ☐ Para administrar o estado global da nossa aplicação.
- ☒ Centralizar as nossas chamadas assíncronas, permitindo-lhes ser mais escaláveis.
- ☐ Centralizar as nossas chamadas síncronas, permitindo-lhes ser mais escaláveis.

Utilizamos o Hook useMutation para: *

0/3

- ☐ Obter informações do servidor.
- ☐ Criar informação no servidor.
- ☒ Atualizar informação no servidor.
- ☐ Apagar informação no servidor.

initialData são dados que persistem em cache, enquanto que placeholderData *3/3 não.

Verdadeiro



Os estados (state) devolvidos por useQuery são: *

0/4

☒ loading

☒ error

☐ paused

☐ success

☐ invalid



Que método falta neste código para limpar o erro quando clicado? *

4/4

```
const CreateTudo = () => {
  const [title, setTitle] = useState('')
  const mutation = useMutation(createTodo)

  const onCreateTudo = e => {
    e.preventDefault()
    mutation.mutate({ title })
  }

  return (
    <form onSubmit={onCreateTudo}>
      {mutation.error && (
        <h5 onClick={() => mutation._____()}>{mutation.error}</h5>
      )}
      <input
        type="text"
        value={title}
        onChange={e => setTitle(e.target.value)}
      />
      <br />
      <button type="submit">Create Todo</button>
    </form>
  )
}
```

- ☐ clear
- ☒ reset
- ☐ resetAll
- ☐ restart

Este formulário foi criado em Digital House.

Google Formulários



