

Iniciado em	sexta, 16 jun 2023, 17:37
Estado	Finalizada
Concluída em	sexta, 16 jun 2023, 18:06
Tempo empregado	29 minutos 8 segundos
Avaliar	6,00 de um máximo de 10,00(60%)



Questão 1

Incorreto Atingiu 0,00 de 1,00

O uso do mecanismo “dotenv” traz, pelo menos, duas vantagens:

- alteração de valores em função do ambiente (desenvolvimento, produção etc)
- isolamento de valores do sistema de controle de versão

O pacote react-dotenv, usado em aula, viabiliza o uso do mecanismo citado em aplicações React. Sabe-se que aplicações React executam puramente do lado do cliente. Diante de tais fatos, analise as seguintes proposições:

I. O arquivo env.js gerado pelo react-dotenv é “baixado” pelo navegador uma vez que a aplicação seja acessada.

II. À chave whitelist, que fica no arquivo package.json, associamos o nome de uma única chave cujo valor associado é de interesse para a aplicação.

III. O arquivo env.js gerado pelo react-dotenv deve ser especificado no arquivo .gitignore e, quando isso é feito, ele deixa de ser “baixado” pelo navegador, uma vez que a aplicação seja acessada.



É correto apenas o que se afirma em

a

- ☐ I
- ☐ II
- ☒ III ✖
- ☐ I e II
- ☐ II e III

Sua resposta está incorreta.

A proposição I é verdadeira. O arquivo env.js é incluído no arquivo index.html e o funcionamento da aplicação depende de sua existência.

A proposição II é falsa. A notação é “whitelist”: [“C1”, “C2”]. Ou seja, à chave whitelist associamos uma coleção.

A proposição III é falsa. Arquivos especificados no arquivo .gitignore deixam de fazer parte do controle de versão apenas. O arquivo env.js criado pelo react-dotenv continua sendo importado pelo arquivo index.html e, portanto, “baixado” pelo navegador.

A resposta correta é:

I.

Questão 2

Correto Atingiu 1,00 de 1,00

Analise as seguintes proposições a respeito do ciclo Redux.

I. Criadoras de ação são objetos Javascript que transitam entre a fase dispatch e os reducers.

II. Um reducer é uma função que, quando acionada, pode decidir não atualizar o estado centralizado.

III. Quando implementa um reducer, o programador está automaticamente implementando um “dispatcher”.

É correto apenas o que se afirma em

- ☐ I
- ☒ II ✓
- ☐ III
- ☐ I e II
- ☐ II e III



Sua resposta está correta.

A proposição I é falsa. Criadoras de ação são, como o nome diz, funções que criam ações. As ações são os objetos que transitam entre a fase de dispatch e os reducers.

A proposição II é verdadeira. De acordo com o type de uma ação recebida, um reducer pode decidir manter a fatia de estado recebida inalterada.

A proposição III é falsa. A fase dispatch nos é entregue pelo próprio redux, ela não é implementada pelo programador.

A resposta correta é:

II.

Questão 3

Correto Atingiu 1,00 de 1,00

Analise a figura e as proposições a seguir.

- I. Sempre que chamada, a função opera sobre uma lista vazia.
- II. Há casos em que ela é executada e a fatia do estado sobre a qual ela opera permanece inalterada.
- III. Ela recebe uma fatia de estado e uma ação.

```
const historicoVestibularReducer = (historicoVestibularAtual = [], acao) =>{  
  if (acao.type === "REALIZAR_VESTIBULAR"){  
    return [...historicoVestibularAtual, acao.payload]  
  }  
  return historicoVestibularAtual  
}
```

É correto apenas o que se afirma em

- ☐ I
- ☐ II
- ☐ III
- ☐ I e II
- ☒ II e III ✓



Sua resposta está correta.

A proposição I é falsa. A função somente opera sobre uma lista vazia caso seu primeiro parâmetro seja undefined. Dentro do ciclo Redux, ele deixa de ser undefined a partir da segunda chamada à função.

A proposição II é verdadeira. Se a condição do if for avaliada como false, a fatia de estado recebida (historicoVestibularAtual) é devolvida sem alterações.

A proposição III é verdadeira. Essa é a definição de um reducer. Seu primeiro parâmetro é a fatia de estado sobre a qual operar e o segundo parâmetro é a ação que representa o que se deseja fazer com a fatia de estado.

A resposta correta é:
II e III.

Questão 4

Incorreto Atingiu 0,00 de 1,00

Considere a figura e as proposições a seguir.

I. Trata-se de um componente React controlado.

II. Caso um props chamado dica não seja especificado quando o componente for utilizado, o campo textual renderizado pelo InputText exibirá "null", sem as aspas.

III. A linha 10 está incorreta já que a função setState deve receber pares chave/valor não englobados por { e }.

```
1  import React, { Component } from 'react'
2  import { InputText } from 'primereact/inputtext'
3  import { Button } from 'primereact/button'
4  export default class Busca extends Component {
5      state = {
6          termoDeBusca: ''
7      }
8      onTermoAlterado = (event) => {
9          console.log(event.target.value)
10         this.setState({ termoDeBusca: event.target.value })
11     }
12     render() {
13         return (
14             // empilhando os filhos
15             <div className="flex flex-column">
16                 /* ícone à esquerda, largura máxima */
17                 <span className="p-input-icon-left w-full">
18                     <i className="pi pi-search" />
19                     <InputText
20                         value={this.state.termoDeBusca}
21                         //largura máxima
22                         className="w-full"
23                         onChange={this.onTermoAlterado}
24                         placeholder={this.props.dica}
25                     />
26                 </span>
27                 <Button
28                     label="OK"
29                     className="p-button-outlined mt-2"
30                 />
31             </div>
32         )
33     }
34 }
35 Busca.defaultProps = {
36     dica: 'Digite algo que deseja ver...'
37 }
38
```



É correto apenas o que se afirma em

- ☐ I
- ☐ II
- ☒ III ✖
- ☐ I e II
- ☐ II e III

Sua resposta está incorreta.

A proposição I é verdadeira. O componente é controlado pois, uma vez que o usuário digite algo, aquilo que ele digitou é armazenado no estado do componente e, a seguir, atribuído ao value do mesmo campo textual.

A proposição II é falsa. O valor exibido será “Digite algo que deseja ver...” graças às linhas de 35 a 37.

A proposição III é falsa. A função setState deve receber um objeto Javascript e a notação correta consiste em englobar os pares chave/valor de interesse com { e }.

A resposta correta é:

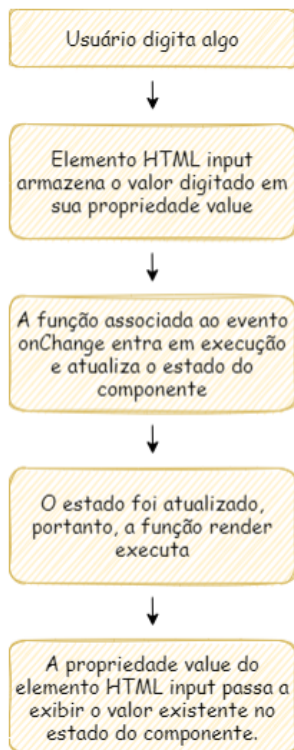
I.



Questão 5

Incorreto Atingiu 0,00 de 1,00

A figura a seguir representa um fluxo de interação com o usuário implementado por um componente React. Ele possui um campo textual (input) em que o usuário digita um valor de interesse. A função associada ao evento onChange entra em funcionamento sempre que o valor do campo textual é atualizado.



Analise as seguintes proposições.

- I. A figura permite concluir que o componente é funcional.
- II. A figura permite concluir que o componente é controlado, desde que ele não possua outros elementos de entrada de dados.
- III. O fluxo representado pela figura não necessariamente inclui o objeto props do componente.

É correto apenas o que se afirma em

- ☐ I
- ☐ II

- ☒ III ✖
- ☐ I e II
- ☐ II e III

Sua resposta está incorreta.

A proposição I é falsa. O fluxo pode ser vivido por componentes funcionais e não funcionais.

A proposição II é verdadeira. Depois de o usuário digitar algo, aquilo que digitou fica armazenado no estado do componente (sob seu controle, daí o nome) e ele mesmo se encarrega de atribuir esse valor explicitamente ao campo value do elemento HTML.

A proposição III é verdadeira. Componentes React podem ou não utilizar o objeto props. Na figura, não há menção a esse objeto.

A resposta correta é:
II e III.



Questão 6

Correto Atingiu 1,00 de 1,00

Quando fazemos a definição de um componente React, desejamos que ele possua características como alto nível de reusabilidade e facilidade de manutenção. Um dos mecanismos relacionados à reusabilidade de um componente React geralmente leva o nome de “props”. A seu respeito, considere as seguintes proposições.

I. Componentes React definidos por meio de classes recebem seu objeto props como um parâmetro do método render.

II. Há componentes React que acessam valores associados a chave(s) disponível(is) em seu objeto props que sequer foram especificadas explicitamente em seu contexto de uso.

III. Embora funções sejam cidadãs de primeira classe em Javascript, elas não podem ser passadas de componente a componente por meio do props.

É correto apenas o que se afirma em

- ☐ I
- ☒ II ✓
- ☐ III
- ☐ I e II
- ☐ II e III



Sua resposta está correta.

A proposição I é falsa. O objeto props é recebido como parâmetro do construtor.

A proposição II é verdadeira. A propriedade children é criada implicitamente pelo React e o componente pode acessá-la com a notação “props.children”.

A proposição III é falsa. Funções podem ser passadas como parâmetros e essa é uma prática muito comum. Veja um exemplo.

```
const Teste = ( ) => {  
  const f = () => console.log('teste')  
  <UmComponenteQualquer funcao=f />  
}
```

A resposta correta é:

II.

Questão 7

Incorreto Atingiu 0,00 de 1,00

Considere o seguinte componente React.

```
1 import React, { Component } from 'react'
2 export default class Q7_estado extends Component {
3   state = {
4     nome: 'João',
5     contador: 0
6   }
7   render() {
8     const atualizar = () => {
9       this.setState({contador: this.state.contador + 1})
10    }
11    return (
12      <div>
13        <p>Olá, {this.state.nome}!</p>
14        <p>Você já clicou {this.state.contador} vezes!</p>
15        <button
16          onClick={atualizar}>
17          OK
18        </button>
19      </div>
20    )
21  }
22 }
```

Analise as seguintes proposições.

- I. A cada clique, a aplicação incrementa o valor armazenado na variável contador e atualiza a tela automaticamente.
- II. A linha 9 substitui o estado original por outro que contém apenas a chave “contador”. Isso quer dizer que, depois do primeiro clique, a variável “nome” passa a valer “undefined” e o nome “João” não é exibido na tela.
- III. A linha 16 pode ser reescrita da seguinte forma, sem que isso altere o funcionamento do componente:

```
onClick={() => atualizar}>
```

É correto apenas o que se afirma em

- ☐ I
- ☐ II
- ☐ III



☒ I e II ✖

☐ II e III

Sua resposta está incorreta.

A proposição I é verdadeira. O vínculo feito na linha 16 está correto. A função atualizar causa a execução da função setState. Ela, por sua vez, atualiza o estado e a tela.

A proposição II é falsa. Embora a linha 9 especifique um valor apenas para a variável contador, as demais chaves são mantidas no estado. Não se trata de uma substituição do objeto inteiro, mas sim de uma atualização dos valores associados às chaves que forem especificadas.

A proposição III é falsa. O código sugerido é uma arrow function que especifica o nome da função atualizar mas não a coloca em execução. Falta a notação () para que isso ocorra. A função vinculada ao evento onClick, portanto, é uma arrow function que não faz nada e devolve uma função. Observe que a função vinculada ao onClick é a arrow function e não a função devolvida por ela.

A resposta correta é:

I.



Questão 8

Correto Atingiu 1,00 de 1,00

Analise as seguintes proposições a respeito do ciclo de vida de um componente React definido por meio de uma classe.

- I. Seu construtor é chamado sempre que seu estado é atualizado.
- II. Seu método render é chamado sempre que seu estado é atualizado.
- III. Os métodos componentDidMount e componentWillUnmount executam uma única vez cada para cada componente instanciado.

É correto apenas o que se afirma em

- ☐ I
- ☐ II
- ☐ III
- ☐ I e II
- ☒ II e III ✓



Sua resposta está correta.

A proposição I é falsa. O construtor é chamado - como o nome sugere - somente no momento em que o componente é construído, o que ocorre uma única vez por componente.

A proposição II é verdadeira. O método render se encarrega de produzir uma nova expressão JSX levando em conta os novos valores existentes no estado atualizado, o que causa a potencial atualização da tela.

A proposição III é verdadeira. O método componentDidMount executa uma única vez, logo depois de o construtor terminar a sua execução e logo após o componente ser montado na árvore DOM. Sua finalidade é permitir que o programador escreva código para a obtenção de recursos necessários para a exibição do componente, de modo que isso seja possível depois de o componente já estar - ainda que parcialmente - visível para o usuário. O método componentWillUnmount executa uma única vez, o que acontece quando ele ainda vai ser desmontado da árvore DOM. Sua finalidade é permitir que o programador escreva código que desaloca recursos eventualmente alocados antes de o componente deixar de existir.

A resposta correta é:
II e III.

Questão 9

Correto Atingiu 1,00 de 1,00

Um desenvolvedor tem como missão desenvolver um componente React capaz de exibir dados de um cosmético. Cosméticos possuem

um título

duas linhas para descrição

um rodapé

Veja a sua primeira tentativa

```
1 import React, { Component } from 'react'
2
3 export default class Q1_componentes extends Component {
4   render() {
5     return (
6       <div>
7         <div>Batom</div>
8         <div>
9           <p>Batom vermelho</p>
10          <p>Coleção tons marcantes</p>
11        </div>
12        <div>
13          Batons XYZ - A melhor marca
14        </div>
15      </div>
16    )
17  }
18 }
19
20
```



Analise as seguintes proposições.

I. Embora possa ser aprimorado do ponto de vista de reusabilidade, o componente é válido. Ou seja, seu uso em uma aplicação não causa erros de compilação ou em tempo de execução.

II. Caso utilize a notação apropriada para o contexto JSX, nada impede que o desenvolvedor especifique regras CSS ao longo da própria definição da função render.

III. Regras CSS somente podem ser especificadas fora da função render e utilizadas por ela caso elas lhe sejam entregues como parâmetros (ou seja, via props).

É correto apenas o que se afirma em

☐ I

- ☐ II
- ☐ III
- ☒ I e II ✓
- ☐ II e III

Sua resposta está correta.

A proposição I é verdadeira. O componente pode ser aprimorado do ponto de vista de reusabilidade pois mostra conteúdo “chumbado” no código. O aprimoramento poderia ser obtido por meio do mecanismo conhecido como props.

A proposição II é verdadeira. Veja um exemplo `<div style={{color: 'blue'}}>`.

A proposição III é falsa. Nada impede que uma regra CSS seja definida fora da função render. Veja: `const regra = {color: 'blue'}`. Ela pode ser usada assim: `<div style={regra}>`. Além disso, a função render não recebe parâmetro algum.

A resposta correta é:
I e II.

Questão 10

Correto Atingiu 1,00 de 1,00

Assinale verdadeiro ou falso. Componentes controlados são aqueles que possuem múltiplos objetos para representar o seu estado.



Escolha uma opção:

- ☐ Verdadeiro
- ☒ Falso ✓

A proposição é falsa. Todo componente React possui um único “estado” que, por sua vez, pode ser composto por múltiplos pares chave/valor.

A resposta correta é 'Falso'.

◀ 2023 - Projeto - Fase 1 - Entrega do grupo + repositório

Seguir para...

2023 - Simulado - P1 ▶

