

Fundamentos de React [24E3_2]

[Página inicial do site](#) ► [Meus cursos](#) ► [Fundamentos de React \[24E3_2\]](#) ► [Etapa 6](#) ► [Teste de Performance - TP3 \[Obrigatório\]](#)

TESTE DE PERFORMANCE - TP3 [OBRIGATÓRIO]

Olá Samuel,

Chegamos em uma das etapas de preparação! A cada Teste de Performance (TP) você terá a oportunidade de praticar os conhecimentos adquiridos e receber feedbacks relevantes para o seu aprendizado.

Tarefa 1

Props

Crie um componente funcional chamado **Greeting** que receba uma prop chamada name e exiba uma saudação utilizando essa propriedade. Por exemplo, se name for "João", o componente deve exibir "Olá, João!".

Instruções:

- Implemente o componente **Greeting** para que ele utilize a prop name e exiba a saudação no formato "Olá, {name}!".
- Utilize o componente **Greeting** em uma aplicação passando diferentes valores para a prop **name**.

Tarefa 2

Ajustando o Estado Inicial

Crie um componente de classe chamado **Counter** que exiba um contador iniciando em 0. Adicione um botão que, quando clicado, incrementa o contador em 1.

Instruções:

- Implemente o componente **Counter** utilizando o estado para controlar o valor do contador.
- Certifique-se de que o valor do contador seja exibido e atualizado conforme o botão é clicado.

Tarefa 3

Atualizando Props

Modifique o componente **Greeting** criado na Questão 1 para aceitar uma prop adicional chamada **age**. O componente deve exibir a saudação e a idade da seguinte forma: "Olá, {name}! Você tem {age} anos."

Instruções:

- Atualize o componente **Greeting** para receber a prop **age**.
- Exiba a saudação com o nome e a idade passados como props no formato "Olá, {name}! Você tem {age} anos."

Tarefa 4

Utilizando Estado em Componentes Filhos

Modifique o componente **Counter** criado na Questão 2 para passar o valor do contador como uma prop chamada **count** para um novo componente funcional chamado **DisplayCount**. Este novo componente deve apenas exibir o valor recebido como prop.

Instruções:

- Crie o componente **DisplayCount** que aceita a prop **count** e exibe seu valor.
- Atualize o componente **Counter** para passar o valor do contador para o componente **DisplayCount** como uma prop **count**.

Tarefa 5 Manipulando o Estado com Funções

No componente **Counter** (implementado na Questão 2), adicione um botão chamado "Reset" que, quando clicado, redefine o contador para 0.

Instruções:

- Adicione uma função ao componente **Counter** que redefine o estado do contador para 0.
- Implemente um botão "Reset" que, ao ser clicado, aciona a função para redefinir o contador para 0.

Tarefa 6 Componentização Avançada

Crie um componente chamado **TodoList** que exibe uma lista de tarefas. Cada tarefa deve ser representada por um componente funcional chamado **TodoItem**, que recebe uma prop chamada **task** contendo o texto da tarefa.

Instruções:

- Implemente os componentes **TodoList** e **TodoItem**, onde **TodoList** renderiza múltiplos **TodoItems** com diferentes tarefas passadas como props.

Tarefa 7 Atualizando Estado de Forma Condicional

Modifique o componente `TodoItem` para adicionar um botão "Concluir" ao lado de cada tarefa. Quando este botão for clicado, a tarefa deve ser marcada como concluída, alterando visualmente sua aparência.

Instruções:

- Adicione a funcionalidade ao botão "Concluir" para atualizar o estado da tarefa no componente `TodoItem`, refletindo visualmente que a tarefa foi concluída.

Tarefa 8 Lidando com Múltiplos Estados

Expanda o componente `TodoList` para permitir adicionar novas tarefas através de um campo de entrada e um botão "Adicionar". Cada nova tarefa adicionada deve ser exibida na lista.

Instruções:

- Implemente a lógica para adicionar novas tarefas ao estado do componente `TodoList` quando o botão "Adicionar" for clicado, e exiba as novas tarefas na lista.

Tarefa 9 Elevando o Estado

Modifique o componente `TodoList` para mover o estado das tarefas para o componente pai que utiliza `TodoList`. As tarefas devem ser passadas como props para o componente `TodoList`.

Instruções:

- Reorganize o código para que o estado das tarefas seja mantido no componente pai e passe as tarefas como props para o componente `TodoList`.

Tarefa 10

Lidando com Eventos Personalizados

Adicione um evento personalizado ao componente `TodoItem` chamado `onTaskDelete`, que é acionado quando um botão "Excluir" ao lado de cada tarefa é clicado.

Instruções:

- Implemente o evento personalizado `onTaskDelete` no componente `TodoItem` e acione-o quando o botão "Excluir" for clicado, passando a tarefa correspondente como argumento.

Tarefa 11

Renderização Condicional

No componente `TodoItem`, adicione a funcionalidade para alternar entre a exibição do texto da tarefa e um campo de entrada para editar a tarefa quando um botão "Editar" é clicado.

Instruções:

- Implemente a renderização condicional no componente `TodoItem` para alternar entre a exibição do texto da tarefa e um campo de entrada quando o botão "Editar" é clicado.
- Garanta que a tarefa seja atualizada conforme o usuário edita o campo de entrada.

Tarefa 12

Renderização Condicional com Elementos em Lista

Crie um componente chamado `List` que recebe uma lista de números como propriedade. Renderize uma lista contendo elementos para cada número. Se a lista estiver vazia, exiba uma mensagem informando que a lista está vazia.

Instruções:

- Implemente o componente **List** para renderizar uma lista contendo elementos para cada número da lista.
- Se a lista estiver vazia, exiba a mensagem "A lista está vazia".

Tarefa 13

Renderização Condicional com Operador Ternário

Crie um componente chamado **Temperature** que exiba "Está quente!" se a temperatura for maior que 25 graus Celsius e "Está frio!" caso contrário.

Instruções:

- Implemente o componente **Temperature** para exibir "Está quente!" se a temperatura passada como prop for maior que 25 graus Celsius.
- Caso contrário, exiba "Está frio!".

Tarefa 14

Props Opcionais

Crie um componente chamado **ProfileCard** que exibe o nome e a idade de um usuário. A propriedade **age** é opcional e, caso não seja fornecida, deve ser exibida uma mensagem indicando que a idade não está disponível.

Instruções:

- Implemente o componente **ProfileCard** para exibir o nome do usuário e sua idade, se fornecida.
- Caso contrário, exiba a mensagem "Idade não disponível".

Tarefa 15

Props Opcionais com Valores Padrão

Crie um componente chamado **Product** que exibe informações sobre um produto, incluindo o nome, preço e uma descrição opcional. Se a descrição não for fornecida, exiba "Descrição não disponível".

Instruções:

- Implemente o componente **Product** para exibir o nome e o preço do produto.
- Se uma descrição for fornecida, exiba-a; caso contrário, exiba "Descrição não disponível".

Tarefa 16

Estilo Condicional

Crie um componente funcional chamado **StatusBadge** que exiba um badge de status baseado em uma prop chamada **status**. O badge deve ter um estilo diferente dependendo do valor da prop **status**.

Instruções:

1. Defina a Prop **status**:

- O componente **StatusBadge** deve aceitar uma prop chamada **status** que pode ter os valores "**ativo**", "**inativo**" ou "**pendente**".

2. Estilos Condicionais:

- Utilize estilos inline ou classes CSS para definir a aparência do badge:
 - Se o **status** for "**ativo**", o badge deve ter um fundo verde e texto branco.
 - Se o **status** for "**inativo**", o badge deve ter um fundo cinza e texto branco.

- Se o **status** for "**pendente**", o badge deve ter um fundo laranja e texto branco.

3. Exibição do Badge:

- O badge deve exibir o texto correspondente ao status. Por exemplo, se o **status** for "**ativo**", o texto do badge deve ser "Ativo".

Assim que terminar, salve todos os links para os exercícios realizados em PDF nomeando o arquivo conforme a regra "nome_sobrenome_DR4_TP3.PDF" e poste como resposta a este TP.

Status da entrega

| | |
|---------------------|-----------------------------|
| Status da entrega | Nenhuma tentativa |
| Status da avaliação | Não avaliado |
| Data de entrega | segunda, 16 set 2024, 23:59 |
| Tempo restante | 7 dias 6 horas |

Template de Rubrica para ser utilizado com a extensão Rubricator

| | | | |
|---|----------------------------------|------------------------------|--|
| 2. Desenvolver componentes em uma interface O aluno executou corretamente 25% dos exercícios da lista proposta? | Não demonstrou o item de rubrica | Demonstrou o item de rubrica | |
| 2. Desenvolver componentes em uma interface O aluno executou corretamente 50% dos exercícios da lista proposta? | Não demonstrou o item de rubrica | Demonstrou o item de rubrica | |
| 2. Desenvolver componentes em uma interface O aluno executou corretamente 75% dos exercícios da lista proposta? | Não demonstrou o item de rubrica | Demonstrou o item de rubrica | |
| 2. Desenvolver componentes em uma interface O aluno executou corretamente 100% dos exercícios da lista proposta? | Não demonstrou o item de rubrica | Demonstrou o item de rubrica | |
| 3. Apresentar dados dinâmicos em uma interface ReactJS O aluno executou corretamente 25% dos exercícios da lista proposta? | Não demonstrou o item de rubrica | Demonstrou o item de rubrica | |
| 3. Apresentar dados dinâmicos em uma interface ReactJS O aluno executou corretamente 50% dos exercícios da lista proposta? | Não demonstrou o item de rubrica | Demonstrou o item de rubrica | |

| | | | |
|--|----------------------------------|------------------------------|--|
| 3. Apresentar dados dinâmicos em uma interface ReactJS O aluno executou corretamente 75% dos exercícios da lista proposta? | Não demonstrou o item de rubrica | Demonstrou o item de rubrica | |
| 3. Apresentar dados dinâmicos em uma interface ReactJS O aluno executou corretamente 100% dos exercícios da lista proposta? | Não demonstrou o item de rubrica | Demonstrou o item de rubrica | |

| | |
|------------------------|-----------------------------------|
| Última modificação | - |
| Comentários da entrega | ▶ Comentários (0) |

ADICIONAR ENTREGA DO TRABALHO

Você ainda não fez um envio.

PÁGINA ANTERIOR

Seguir para... ▾

PRÓXIMO PÁGINA

SITES DAS ESCOLAS

[Faculdade Infnet](#)

LINKS RÁPIDOS

[Requerimentos](#)

ATENDIMENTO NO

TEAMS

REDES SOCIAIS

