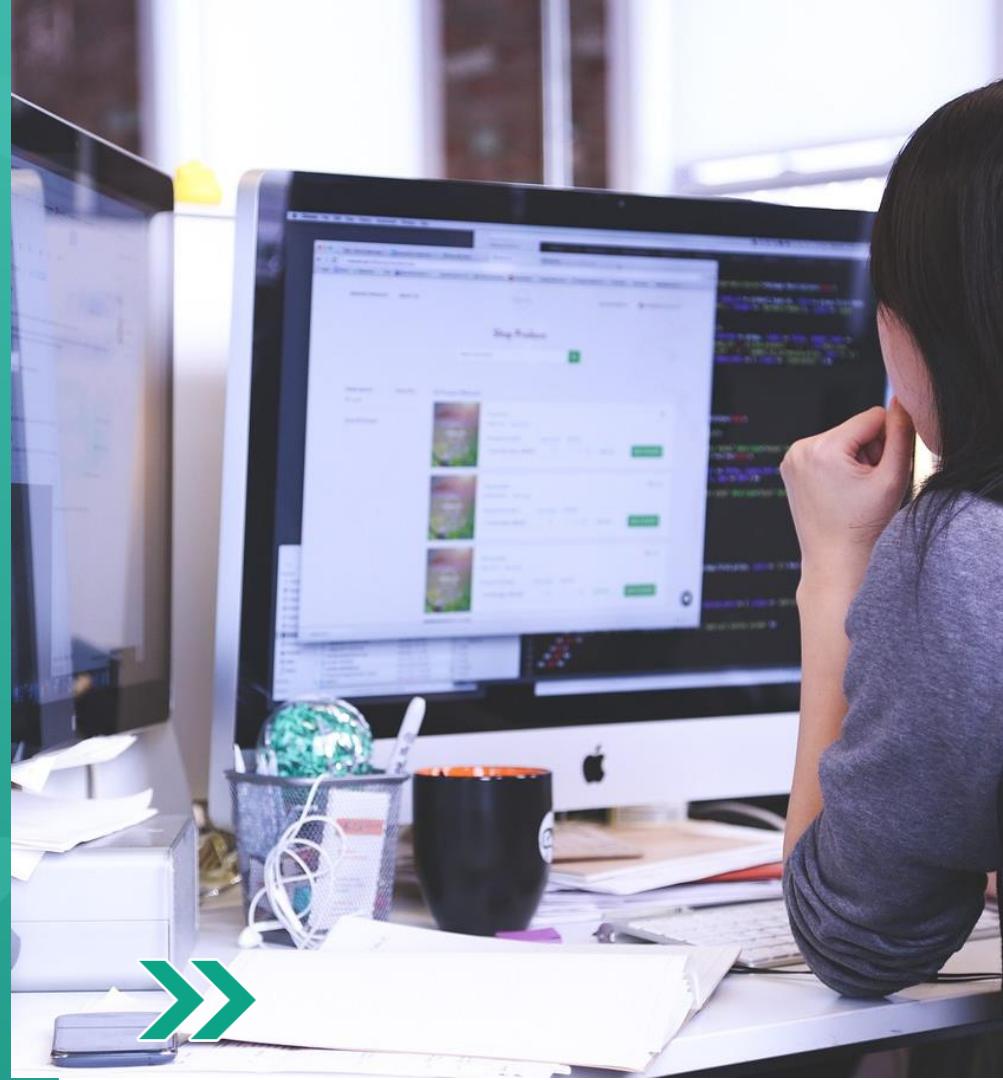


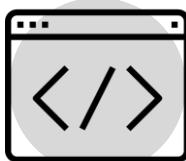
# Profissão: Engenheiro Front-End



# BOAS PRÁTICAS



# Projeto 5 – Continue o To Do List



Confira boas práticas da  
comunidade de Front-End por  
assunto relacionado às aulas.

- **Remova uma tarefa**
- **Edite uma tarefa**
- **Conheça React  
Router DOM**
- **Navegue entre  
rotas**
- **Crie uma tarefa**



# Remova uma tarefa



## Dispatch

Existem vários métodos diferentes de despacho, incluindo despacho estático e despacho dinâmico. No despacho estático, o método a ser executado é determinado em tempo de compilação com base no tipo do objeto ou dos argumentos. No despacho dinâmico, o método a ser executado é determinado em tempo de execução com base no valor real do objeto ou dos argumentos.

# Edite uma tarefa

Acompanhe algumas dicas para usar componentes controláveis.



## **Gerencie o estado do componente:**

Certifique-se de que o estado do componente controlável seja gerenciado adequadamente. Use o useState ou um gerenciador de estado apropriado para criar e atualizar o estado do componente quando ocorrerem mudanças nos elementos de formulário.



## **Inicialize o estado corretamente:**

Ao criar um componente controlável, certifique-se de inicializar o estado com um valor adequado para o elemento de formulário. Por exemplo, se for um input de texto, o estado inicial pode ser uma string vazia ou algum valor padrão.



## **Use o valor do estado para definir o valor do elemento de formulário:**

O valor do estado do componente deve ser utilizado para definir o valor dos elementos de formulário controlados. Por exemplo, ao usar um input controlado, defina o atributo value do input com o valor do estado.

# Edite uma tarefa

Acompanhe algumas dicas para usar componentes controláveis.



## Valide os dados do formulário:

Ao usar componentes controláveis, é mais fácil validar os dados do formulário, já que o estado contém o valor mais recente do elemento de formulário. Implemente a lógica de validação dentro da função `onChange` ou antes de enviar os dados do formulário.



## Combine componentes controlados com eventos de submissão:

Ao trabalhar com formulários, é comum utilizar eventos de submissão (por exemplo, `onSubmit` em um elemento `<form>`). Combine componentes controlados com eventos de submissão para enviar os dados do formulário com base no estado atualizado dos componentes.



## Considere o desempenho:

Em alguns casos, componentes controlados podem causar re-renderizações frequentes, especialmente em formulários grandes. Se você notar problemas de desempenho, considere usar técnicas como `useCallback` para evitar recriações desnecessárias de funções de evento.

# Conheça React Router DOM

Acompanhe algumas dicas para usar o React Router.



## Defina suas rotas:

Defina as rotas da sua aplicação usando o componente Route. Cada Route deve ter uma propriedade path que corresponda à URL desejada e uma propriedade component que indique o componente React a ser renderizado quando a URL corresponder à rota.



## Aninhe suas rotas:

Se sua aplicação possui hierarquia de rotas, você pode aninhar Route dentro de outros componentes para criar rotas aninhadas. Isso permite que você renderize componentes específicos dentro de outros componentes baseados nas URLs correspondentes.



## Link para navegação:

Utilize o componente Link para criar links de navegação em sua aplicação. O Link renderiza âncoras `<a>` que evitam recarregar a página inteira ao navegar entre as rotas, garantindo uma experiência de usuário mais suave.

# Conheça React Router DOM

Acompanhe algumas dicas para usar o React Router.



## Parâmetros de rota:

Use parâmetros de rota para tornar suas rotas mais dinâmicas e flexíveis. Você pode definir parâmetros na propriedade path de uma Route e, em seguida, acessá-los em seus componentes usando useParams (para componentes funcionais) ou match.params (para componentes de classe).



## Histórico de Navegação:

Você pode acessar o histórico de navegação da aplicação usando o hook useHistory. Isso permite que você programe ações de navegação, como voltar, avançar ou navegar para uma rota específica programaticamente.



## Redirecionamento:

Utilize o componente Redirect para redirecionar os usuários para outras rotas quando eles acessarem uma URL específica ou realizar alguma ação específica em sua aplicação.

# Navegue entre rotas

Acompanhe algumas dicas para usar o componente Link.



## Substituição de âncoras `<a>`:

O componente Link é usado para criar links de navegação em sua aplicação, e ele substitui o uso das âncoras `<a>` para navegar entre as diferentes rotas. Utilizar o Link ao invés do `<a>` garante que a navegação ocorra de forma suave, sem a necessidade de recarregar a página inteira.



## Evitar o comportamento padrão:

Ao usar o Link, você não precisa adicionar eventos de clique e chamar o método `event.preventDefault()` para evitar que a página seja recarregada. O Link cuida disso automaticamente, fornecendo a navegação SPA (Single Page Application) sem esforço adicional.



## Definir a propriedade "to":

O Link requer a propriedade "to" para especificar a URL para onde o usuário será direcionado quando o link for clicado. Essa propriedade deve conter o caminho para a rota desejada.

# Navegue entre rotas

Acompanhe algumas dicas para usar o componente Link.



## Usar caminhos relativos ou absolutos:

A propriedade "to" do Link pode receber caminhos relativos (ex: "/about") ou caminhos absolutos (ex: "https://exemplo.com/sobre"). Use caminhos relativos quando estiver navegando dentro da aplicação e caminhos absolutos para redirecionar para outras páginas externas.



## Complemente com outras rotas:

O Link é mais útil quando usado em conjunto com rotas definidas pelo componente Route. Ao criar uma hierarquia de rotas, os links serão correspondentes aos caminhos definidos nas rotas, garantindo a consistência na navegação.



## Passagem de parâmetros:

Se suas rotas requerem parâmetros, você pode passá-los através da propriedade "to" usando template strings. Por exemplo, para passar um ID como parâmetro: <Link to={`/detalhes/\${id}`}>.

# Navegue entre rotas

Acompanhe algumas dicas para usar o componente Link.



## Utilização em outros componentes:

O Link pode ser usado em qualquer componente que esteja dentro da árvore de roteamento do React Router. Isso inclui componentes funcionais, componentes de classe e até mesmo dentro do conteúdo de um componente.

# Crie uma tarefa



## UseNavigate:

Essa função também aceita um objeto como parâmetro, que pode ser usado para passar parâmetros para a nova rota.

# Bons estudos!

