



# Pensando do jeito React

- Antes de construirmos aplicações em React, primeiro temos que quebrar nossa interface (user interface) em pequenas partes, que chamamos de **componentes**;

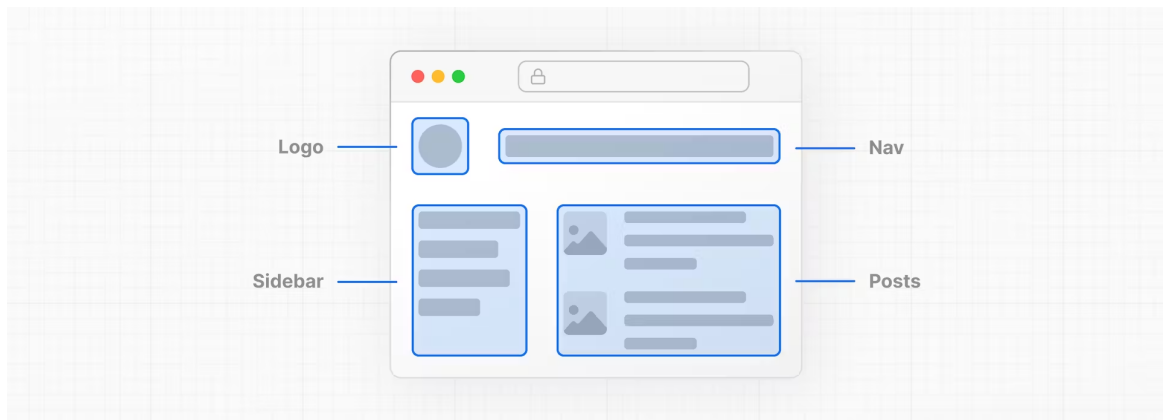


A **User Interface (UI)**, ou **Interface do Usuário** em português, refere-se à forma como nós, os usuários, interagimos com um software, aplicativo ou sistema. Inclui tudo o que vemos na tela, como botões, cores, desenhos e até mesmo como podemos interagir com esses elementos (touch ou click do mouse, por exemplo).

- Imagine que temos que quebrar uma tela da nossa aplicação em pequenos pedaços, como se fossem peças de um quebra-cabeças;
  - Cada peça desse quebra-cabeças será o nosso componente e, ao juntarmos tudo, teremos a nossa aplicação final, na qual os usuários irão interagir com o nosso produto.

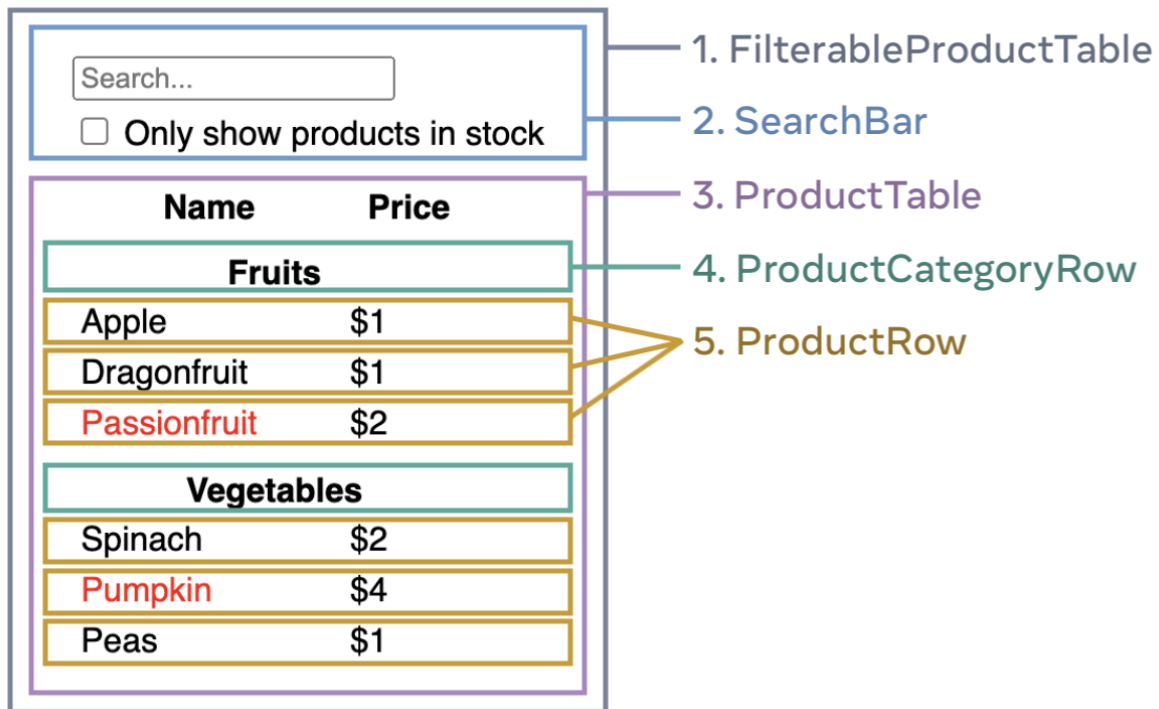
## Quebrando nosso design em componentes

- Pegue o design de cada tela da sua aplicação faça o exercício de quebrá-lo em pequenas partes/componentes;
- Assim, conseguimos visualizar a estrutura na qual o nosso código terá para que nós possamos começar a desenvolver nossa aplicação React:



Exemplo de divisão dos componentes da aplicação

- Vale lembrar que podemos ter componentes e sub-componentes dentro dessa estrutura e, desenhar quadrados em volta, separando cada componente visualmente, vai nos ajudar no momento de construir nossa aplicação e de visualizar;
- Uma boa forma de dividirmos os nossos componentes, é pensarmos se aquele “bloco” de código irá ter uma responsabilidade única, ou seja, ele irá ter apenas **um propósito**, ele irá cuidar de exibir ou trabalhar com apenas **uma função** dentro da nossa aplicação;
  - Se esse componente acabar crescendo ao longo do tempo e tendo mais de uma função, o ideal é nós quebrarmos ele em subcomponentes menores.
- Outro ponto importante é pensarmos nos componentes como um bloco que pode ser **reutilizado**, ou seja, repetido várias vezes conforme a necessidade, tendo a mesma estrutura, porém, podendo estar exibindo informações diferentes.



No exemplo acima, a divisão do componente ficou:

- `FilterableProductTable` (cinza): contém toda a nossa aplicação/tela;
  - `SearchBar` (azul): Recebe as entradas de informações do usuário, como o termo de busca que ele gostaria de utilizar e se ele gostaria de buscar apenas produtos em estoque;
  - `ProductTable` (roxo): Exibe e filtra a lista de acordo com o que o usuário adicionou no filtro de busca do componente `SearchBar`;
  - `ProductCategoryRow` (verde): Exibe um texto de cabeçalho para cada categoria da listagem;
  - `ProductRow` (amarelo): Se refere à linha de cada produto da listagem.
- Devemos lembrar que não há certo ou errado no momento de realizarmos a divisão dos componentes da nossa aplicação. O importante é seguirmos a

premissa de termos componentes que tenham uma única responsabilidade, com o intuito de deixar a manutenção do nosso código mais simples, rápida e fácil;

- No exemplo acima, temos definida a hierarquia dos componentes da seguinte forma:
  - `FilterableProductTable`
    - `SearchBar`
    - `ProductTable`
      - `ProductCategoryRow`
      - `ProductRow`

## Passo 2: Criando uma versão estática em React

- A forma mais simples de criarmos uma aplicação React, depois de fazermos o exercício de quebrar nossa página em componentes, é programarmos esses componentes de forma estática, ou seja, sem implementarmos interatividade;
- Aqui iremos apenas implementar recursos como o HTML e CSS dos nossos componentes, sem utilizar outros recursos do React que veremos no futuro, como o uso de estados, por exemplo.

## Passo 3: Adicionando interações aos componentes


- O próximo passo é adicionarmos as interações dos usuários com os componentes que criamos. Aqui, iremos utilizar mais raciocínio lógico para finalizarmos a nossa aplicação.
  - Nesse exemplo, podemos fazer com que os usuários sejam capazes de filtrar a listagem de frutas e vegetais;
  - Se houvessem outras interações, como selecionar os itens e adicionar à um carrinho de compras, por exemplo, também faríamos nesse momento;
  - Pode ser aqui também o momento de realizarmos a junção entre os nossos componentes com o uso de uma API, dependendo da necessidade da aplicação.

⚠ Vale lembrar que essa ordem é apenas uma sugestão de como podemos pensar em um projeto e codificarmos esse produto utilizando o React. Conforme o tempo for passando, você irá entender e construir sua aplicação da forma que achar mais eficiente e que funcionar melhor para você.

## Referências

Thinking in React – React

The library for web and native user interfaces

 <https://react.dev/learn/thinking-in-react>

 React

Learn React

REACT.DEV/LEARN