

## 1. useState

- **Descrição:** Hook que adiciona estado local a componentes funcionais.

### Sintaxe:

javascript

Copiar código

```
const [state, setState] = useState(initialState);
```

- **Uso no Código:**
  - **city:** Armazena o nome da cidade digitada pelo usuário.
  - **setCity:** Função para atualizar o estado de **city**.
  - **weatherInfo:** Armazena as informações da previsão do tempo.
  - **setWeatherInfo:** Função para atualizar o estado de **weatherInfo**.
  - **error:** Armazena mensagens de erro, como "Cidade não encontrada".
  - **setError:** Função para atualizar o estado de **error**.

## 2. useEffect

- **Descrição:** Hook para executar efeitos colaterais em componentes funcionais, como chamadas de API ou manipulação de DOM.

### Sintaxe:

javascript

Copiar código

```
useEffect(() => {  
  // Código do efeito colateral  
  return () => {  
    // Código de limpeza (opcional)  
  };  
}, [dependencies]);
```

- 

- **Uso no Código:**
  - Não utilizado explicitamente no código fornecido, mas geralmente usado para realizar ações após renderizações e limpar efeitos quando o componente é desmontado ou quando as dependências mudam.

Esses hooks são essenciais para gerenciar estados e efeitos em componentes funcionais do React, permitindo uma abordagem mais declarativa e modular para criar interfaces de usuário interativas