

# Componente React de Visualização em Árvore

## Visualização em árvore

Um modo de visualização em árvore apresentando uma lista hierárquica.

As visualizações em árvore podem ser usadas para representar um navegação no sistema de arquivos para exibir pastas e arquivos, um item representando uma pasta pode ser expandido para revelar o conteúdo da pasta, que pode ser arquivos, pastas ou ambos.

```
{{"component": "modules/components/ComponentLinkHeader.js"}}
```

## Modo básico de visualização em árvore

```
{{"demo": "FileSystemNavigator.js"}}
```

## Seleção múltipla

Visualizações de árvore também suportam seleção múltipla.

```
{{"demo": "MultiSelectTreeView.js"}}
```

## Visualização em árvore controlada

A visualização em árvore também oferece uma API para controle.

```
{{"demo": "ControlledTreeView.js"}}
```

## Objeto complexo

Enquanto o componente `TreeView/TreeItem` maximiza a flexibilidade, um passo extra é necessário para lidar com um objetos complexos.

Vamos considerar uma variável de dados com a seguinte estrutura, a recursão pode ser usada para lidar com este cenário.

```
const data = {  
  id: 'root',  
  name: 'Parent',  
  children: [  
    {  
      id: 'child1',  
      name: 'Child 1',  
      children: [  
        {  
          id: 'grandchild1',  
          name: 'Grandchild 1',  
          children: []  
        },  
        {  
          id: 'grandchild2',  
          name: 'Grandchild 2',  
          children: []  
        }  
      ]  
    },  
    {  
      id: 'child2',  
      name: 'Child 2',  
      children: []  
    }  
  ]  
}
```

```

    {
      id: '1',
      name: 'Child - 1',
    },
    // ...
  ],
};
{{"demo": "RichObjectTreeView.js", "defaultCodeOpen": false}}
```

## Propriedade ContentComponent

Você pode usar a propriedade `ContentComponent` e o hook `useTreeItem` para customizar ainda mais o comportamento do `TreeItem`.

Como limitar a expansão para clicar no ícone:

```
{{"demo": "IconExpansionTreeView.js", "defaultCodeOpen": false}}
```

Ou aumentando a largura do indicador de estado:

```
{{"demo": "BarTreeView.js", "defaultCodeOpen": false}}
```

## Visualização em árvore customizada

### Ícones customizados, borda e animação

```
{{"demo": "CustomizedTreeView.js"}}
```

### Clone do Gmail

```
{{"demo": "GmailTreeView.js"}}
```

## Itens desabilitados na árvore

```
{{"demo": "DisabledTreeItems.js"}}
```

O comportamento dos itens desabilitados da árvore depende da propriedade `disabledItemsFocusable`.

Se é falsa:

- As teclas de setas não focarão nos itens desabilitados e o próximo item não desabilitado será focado.
- Digitar o primeiro caractere do rótulo de um item desabilitado não focará no item.
- Interação do mouse ou teclado não irá expandir/recolher itens desabilitados.
- Interação do mouse ou teclado não selecionará itens desabilitados.
- Shift + teclas de setas irão pular itens desabilitados e o próximo item não desabilitado será selecionado.
- Foco programático não focará itens desabilitados.

Se é verdadeira:

- As teclas de setas focarão em itens desabilitados.
- Digitar o primeiro caractere do rótulo de um item desabilitado focará no item.
- Interação do mouse ou teclado não irá expandir/recolher itens desabilitados.
- Interação do mouse ou teclado não selecionará itens desabilitados.
- Shift + teclas de setas não irão pular itens desabilitados, mas o item desabilitado não será selecionado.
- Foco programático focará itens desabilitados.

## Acessibilidade

(WAI-ARIA: <https://www.w3.org/TR/wai-aria-practices/#TreeView>)

O componente segue as práticas de autoria da WAI-ARIA.

Para ter uma exibição em árvore acessível, você deve usar `aria-labelledby` ou `aria-label` para fazer referência ou fornecer um rótulo na `TreeView`, caso contrário, os leitores de tela irão anunciá-lo como “tree”, tornando difícil entender o contexto de um item específico da árvore.