**React Beautiful DND**

<https://github.com/atlassian/react-beautiful-dnd>

Criação de um novo projeto em React:

* npx create-react-app task-app

Adição da biblioteca de drag and drop:

* npm add react-beatuful.dnd

**Componentes da biblioteca:**

**Uma imagem contendo placa, rua, amarelo, tráfego

Descrição gerada automaticamente**

* **<DragDropContext />** - Agrupa a parte do seu aplicativo que você deseja ativar e arrastar para.
  + Props obrigatórios:
    - onDragEnd : chamado no fim do araste
  + Props opcionais:
    - onDragUpdate: chamado quando algo muda durante o arraste
    - onDragEnd: chamado no início do arraste
* **<Droppable />** - Área em que pode solto o componente arrastado. Contém <Draggable /> s dentro dele.
  + Props obrigatórios:
    - droppableId
  + Provided (DroppableProvided):
    - innerRef: (?HTMLElement) => void – elemento do DOM
    - droppableProps: DroppableProps - Este é um objeto que contém propriedades que precisam ser aplicadas a um elemento solto na tela.
    - placeholder: ?Node - É usado para criar espaço no <Droppable /> conforme necessário durante um arrasto
  + Exemplo:

<Droppable droppableId="droppable-1">

    {(provided, snapshot) => (

        <div ref={provided.innerRef} {...provided.droppableProps}>

            Good to go(Aqui deve ter um elemento <Draggable />)

            {provided.placeholder}

        </div>

    )}

</Droppable>

* **<Draggable />** - O que pode ser arrastado.
  + Props obrigatórios:
    - draggableId
    - índice - Um requerido number que corresponda à ordem do <Draggable />no <Droppable />. É simplesmente o índice da <Draggable />na lista.
  + Provided (DraggableProvided):
    - innerRef: (?HTMLElement) => void – elemento do DOM
    - draggableProps : DraggableProps - é um objeto que contém um dataatributo e uma linha style. Controla o movimento do arrastável quando está arrastando e não arrastando.
    - dragHandleProps :? DragHandleProps - É usado para arrastar o todo <Draggable />
  + Exemplo:

<Draggable draggableId={this.props.task.id} index={this.props.index}>

    {(provided, snapshot) => (

        <Container

            {...provided.draggableProps}

            {...provided.dragHandleProps}

            ref={provided.innerRef}

            isDragging={snapshot.isDragging}

        >

            {this.props.task.content}

        </Container>

    )}

</Draggable>

* + Observações:

Outra propriedade provida pelo <Draggable> é o **snapshot**. Este objeto pode ser usado para dar style ao componente arrastado.

* **resetServerContext()** - função deve ser usada quando a renderização do lado do servidor (SSR). Ele garante que o estado do contexto não persista em várias renderizações no servidor, o que resultaria em incompatibilidades de marcação de cliente / servidor depois que várias solicitações forem renderizadas no servidor.

**Referências:**

<https://egghead.io/lessons/react-conditionally-allow-movement-using-react-beautiful-dnd-draggable-and-droppable-props> - video aulas explicando como montar um projeto do zero com a biblioteca

<https://github.com/atlassian/react-beautiful-dnd> - github da biblioteca

<https://medium.com/@yudhajitadhikary/implementation-of-drag-drop-using-react-beautiful-dnd-and-material-ui-465be23b1bd9> - documentação e informações da biblioteca

<https://github.com/tatianaValente/task-app> - repositório no github da Tatiana com implementação do drag and drop