# Documentação do Projeto “Dynamics Components”

## **Descrição do Projeto**

O projeto Dynamics Components foi desenvolvido utilizando React para praticar a criação e utilização de componentes. O objetivo é consolidar o entendimento sobre a troca de dados via propriedades, o uso do atributo children e a construção de interfaces dinâmicas com formulários e listas.

## **Estrutura do Projeto**

O projeto é composto pelos seguintes componentes:

1. ****App**:** Componente principal que gerencia os demais componentes e suas interações.
2. ****FooterComponent**:** Componente que exibe o rodapé da aplicação, incluindo direitos autorais.
3. ****FormComponent**:** Componente responsável por renderizar um formulário que coleta dados do usuário.
4. ****ListComponent:**** Componente que exibe uma lista de itens com base nos dados fornecidos pelo usuário.
5. ****ListItemComponent**:** Componente que representa um item individual na lista.
6. ****AnotherComponent**:** Componente adicional que demonstra a utilização do atributo children.
7. ****ApiComponent**:** Componente que busca dados de uma API e os exibe em uma lista.

## **Comunicação entre Componentes**

Os componentes se comunicam através de props. O FormComponent envia dados para o ListComponent, que, por sua vez, utiliza o ListItemComponent para renderizar cada item da lista.

**Exemplo de troca de dados via props:**

<FormComponent onSubmit={this.handleAddItem} />

<ListComponent items={this.state.items} />

## **Uso do Atributo children**

O AnotherComponent demonstra o uso do atributo children para renderizar conteúdo dinâmico.

**Exemplo:**

<AnotherComponent>

<p>Este é um conteúdo dinâmico!</p>

</AnotherComponent>

## **Funcionalidade do Formulário**

O formulário no FormComponent permite que os usuários insiram novos itens, que são adicionados à lista exibida pelo ListComponent.

**Exemplo de implementação do formulário:**

<form onSubmit={this.handleSubmit}>

<input type="text" value={this.state.inputValue} onChange={this.handleInputChange} />

<button type="submit">Adicionar</button>

</form>

## **Listagem de Itens**

O ListComponent renderiza a lista de itens, utilizando o ListItemComponent para cada item da lista.

**Exemplo de listagem:**

<ul>

{this.props.items.map(item => (

<ListItemComponent key={item.id} item={item} />

))}

</ul>

## **Instruções de Uso**

1. **Instalação:** Clone o repositório e execute **npm install** para instalar as dependências.
2. **Execução:** Utilize **npm start** para iniciar o servidor de desenvolvimento e acessar a aplicação em [http://localhost:3000](http://localhost:3000/).

## **Conclusão**

O projeto Dynamics Components foi desenvolvido com foco nas melhores práticas de React, incluindo a modularização do código, documentação e organização do repositório. Siga as instruções acima para replicar e entender o funcionamento da aplicação.