

TodoList Frontend React

Criando o Componente de Entrada de Dados

Crie na pasta componentes, um arquivo chamado TodoForm.jsx com o conteúdo abaixo:

```
import React from 'react'

export const TodoForm = () => {
  return (
    <form className="TodoForm">
      <input type="text" className="todo-input" placeholder='Descrição da Tarefa' />
      <button type="submit" className='todo-btn'>Adicionar Tarefa</button>
    </form>
  )
}
```

Incluindo o componente de Entrada de Dados no Componente Principal

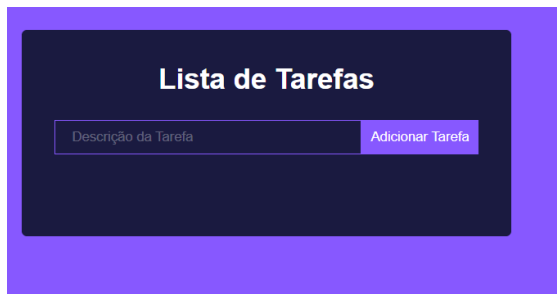
Edite o componente TodoWrapper, incluindo a chamada ao componente TodoForm, criado no item acima, conforme o exemplo abaixo:

```
import React from 'react';
import { TodoForm } from './TodoForm';

export const TodoWrapper = () => {
  return (
    <div className='TodoWrapper'>
      <h1>Lista de Tarefas</h1>
      <TodoForm />
    </div>
  );
}
```

Verificando os resultados

Agora deveremos ter nossa tela, com a opção de inclusão de uma tarefa, porém, ainda não funcional, conforme o exemplo abaixo:



Criando o Componente que Lista as Tarefas

Crie na pasta componentes, um arquivo chamado `TodoList.jsx` com o conteúdo abaixo:

```
import React from 'react'

export const TodoList = ({task}) => {
  return (
    <div className="Todo">
      <p className={` ${task.completed ? "completed" : "incompleted"} `}>
        >{task.description}</p>
      </div>
    )
  }
}
```

Incluindo o componente que Lista as Tarefas no Componente Principal

Edite o componente `TodoWrapper`, incluindo a chamada ao componente `TodoForm`, criado no item acima, conforme o exemplo abaixo:

```
import React, { useState } from 'react';
import { TodoForm } from './TodoForm';
import { TodoList } from './TodoList';

export const TodoWrapper = () => {
  const [todos, setTodos] = useState([
    {id: 1, description: 'Tarefa exemplo', completed: false}
  ]);

  return (
    <div className='TodoWrapper'>
      <h1>Lista de Tarefas</h1>
      <TodoForm />
      {todos.map((item) =>
        <TodoList
          key={item.id}
          task={item}
        />
      )}
    </div>
  )
}
```

```
);  
}
```

Verificando os resultados

Agora deveremos ter nossa tela, com a opção de inclusão de uma tarefa e com a lista de itens padrões, porém, ainda não funcional, conforme o exemplo abaixo:



Criando o metodo addTodo no TodoWrapper

Vamos editar o componente TodoWrapper, incluindo as ações necessárias, para a inclusão de uma nova tarefa (addTodo), conforme o exemplo abaixo:

```
import React, { useState } from 'react';  
import { TodoForm } from './TodoForm';  
import { TodoList } from './TodoList';  
import { v4 as uuidv4 } from "uuid";  
  
export const TodoWrapper = () => {  
  const [todos, setTodos] = useState([  
    {id: 1, description: 'Tarefa exemplo', completed: false}  
  ]);  
  
  const addTodo = (todo) => {  
    setTodos([  
      ...todos,  
      { id: uuidv4(), task: description, completed: false },  
    ]);  
  }  
  
  return (  

```

```

<div className='TodoWrapper'>
  <h1>Lista de Tarefas</h1>
  <TodoForm addTodo={addTodo}/>
  {todos.map((item) =>
    <TodoList
      key={item.id}
      task={item}
    />
  )
}
</div>
);
}

```

Preparando addTodo no todoForm

Vamos editar o componente TodoForm, incluindo as ações necessárias, para a inclusão de uma nova tarefa (addTodo), conforme o exemplo abaixo:

```

import React, {useState} from 'react'

export const TodoForm = ({addTodo}) => {
  const [value, setValue] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    if (value) {
      // adicionar tarefa
      addTodo(value);
      // limpar formulário apos envio
      setValue('');
    }
  };

  return (
    <form className="TodoForm" onSubmit={handleSubmit} >
      <input type="text" value={value} onChange={(e) =>
setValue(e.target.value)} className="todo-input" placeholder='Descrição da
Tarefa' />
      <button type="submit" className='todo-btn'>Adicionar Tarefa</button>
    </form>
  )
}

```

Criando o Componente Service

Crie na pasta componentes, um arquivo chamado `TodoWrapperService.jsx`. Este componente será responsável por fazer a lógica da nossa aplicação. Inicialmente vamos trabalhar apenas com os dados locais, para entender o funcionamento e depois integramos com o nosso backend.

```
import React, {useState, useEffect} from 'react'

export const TodoWrapperService = () => {
  const [todos, setTodos] = useState([])

  useEffect(() => {
    const savedTodos = JSON.parse(localStorage.getItem('todos')) || [];
    setTodos(savedTodos);
  }, []);

  return (
    <div className='TodoWrapper'>
      <h1>Lista de Tarefas! (Service)</h1>
      <TodoForm />
      {todos.map((todo, index) => (
        <TodoList task={todo} key={index} />
      ))}
    </div>
  )
}
```

Verificando os resultados

Agora deveremos ter nossa tela, com a opção de inclusão de uma tarefa e com a lista de itens funcionando a inclusão de novos itens, conforme o exemplo abaixo:



The screenshot shows a web application titled "Lista de Tarefas". The interface is dark-themed with a light blue border. At the top, the title "Lista de Tarefas" is displayed in large white font. Below the title, there is a form with a light blue input field containing the text "teste" and a blue button labeled "Adicionar Tarefa".

