

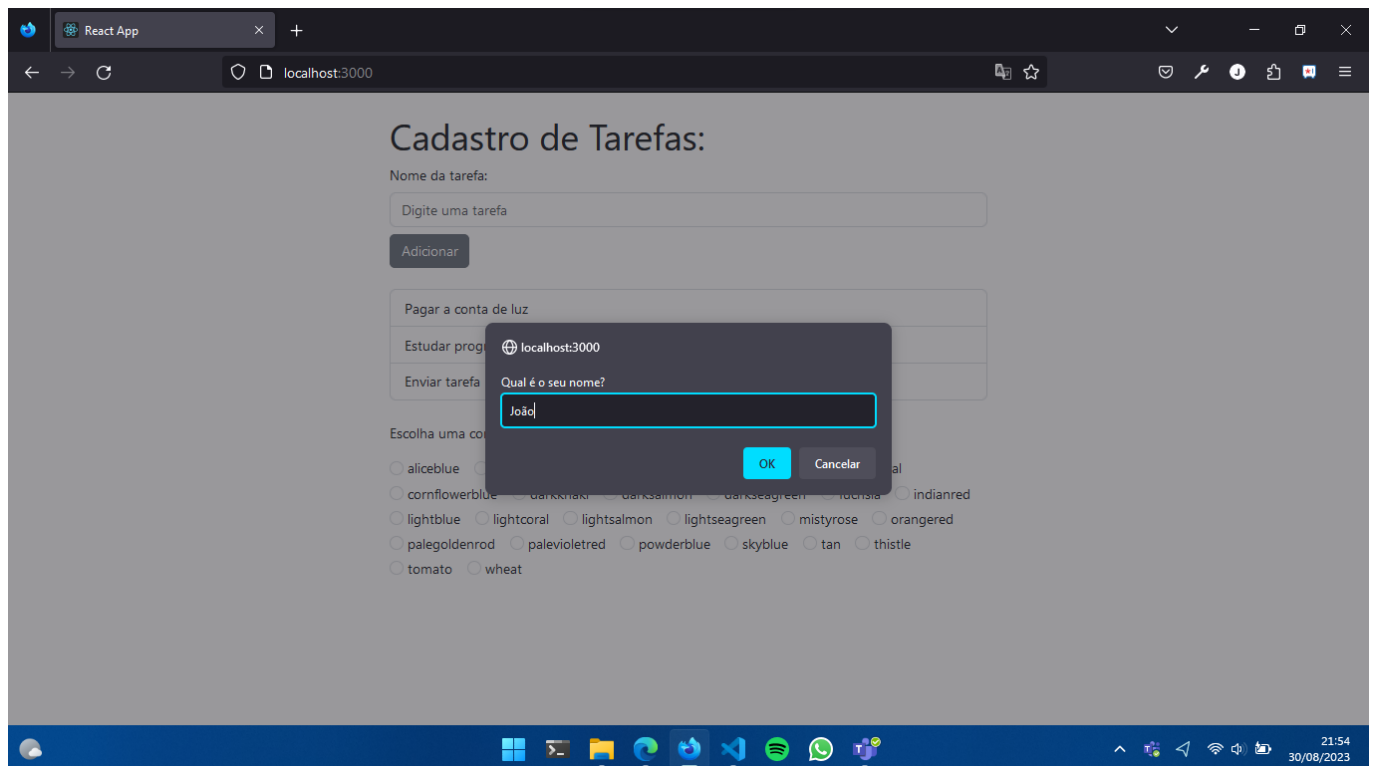
[Tarefa 2] Trabalhando com Arrays, Hooks e localStorage

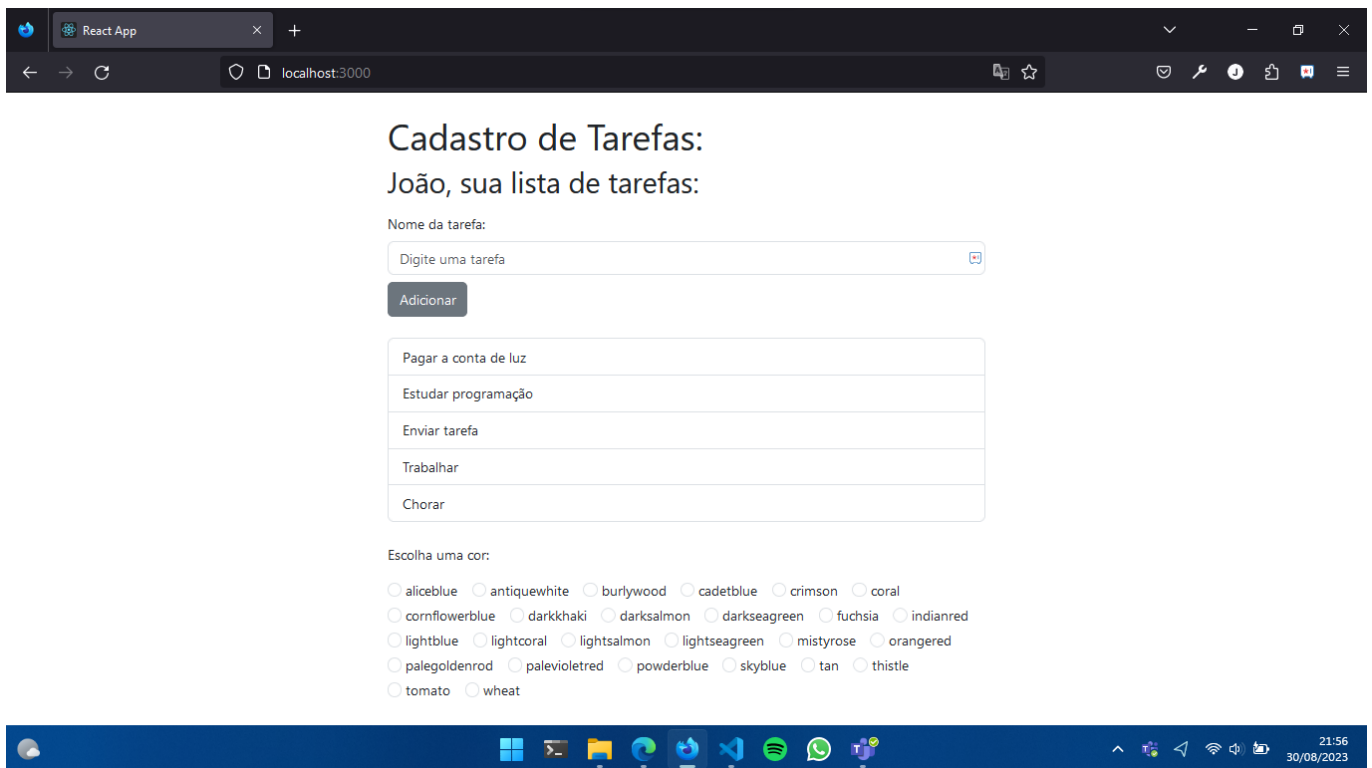
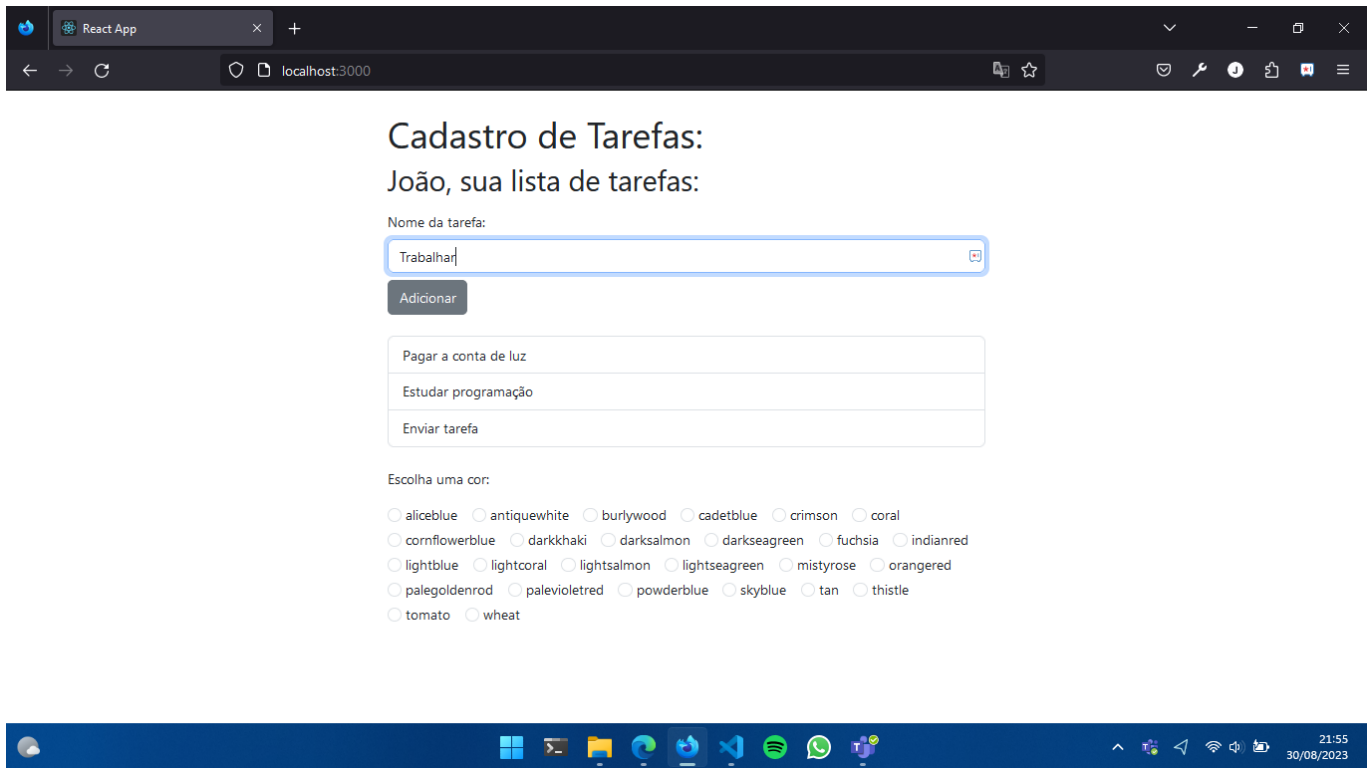
Instruções

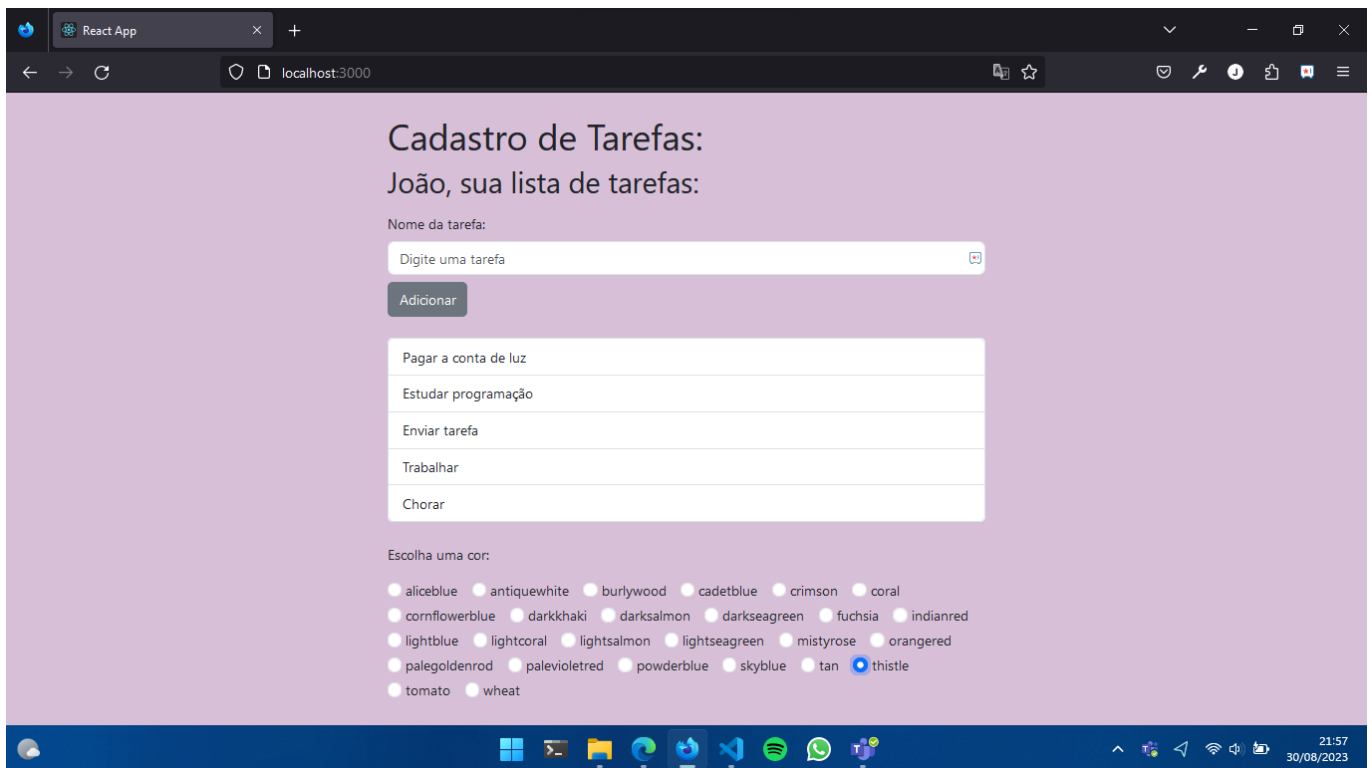
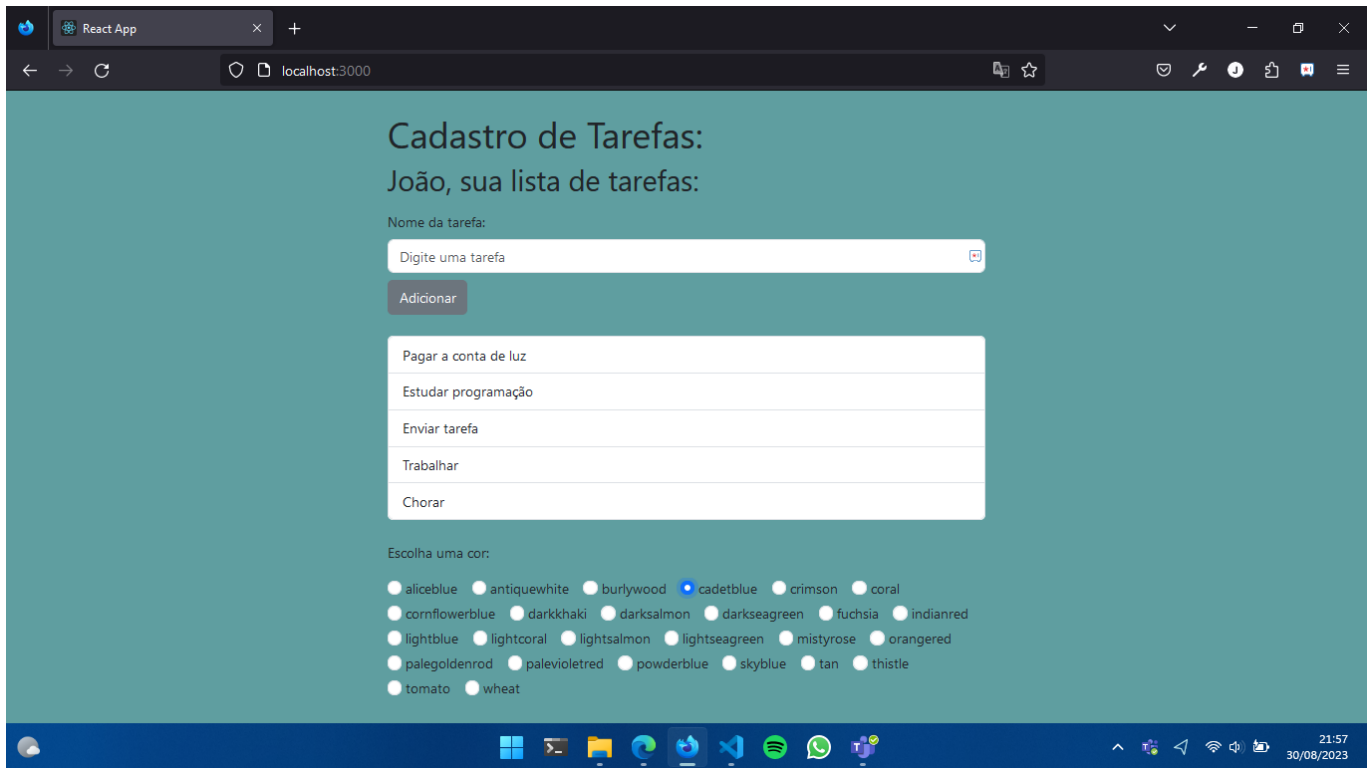
Utilizando o mesmo projeto da Aula 1, faça as seguintes mudanças:

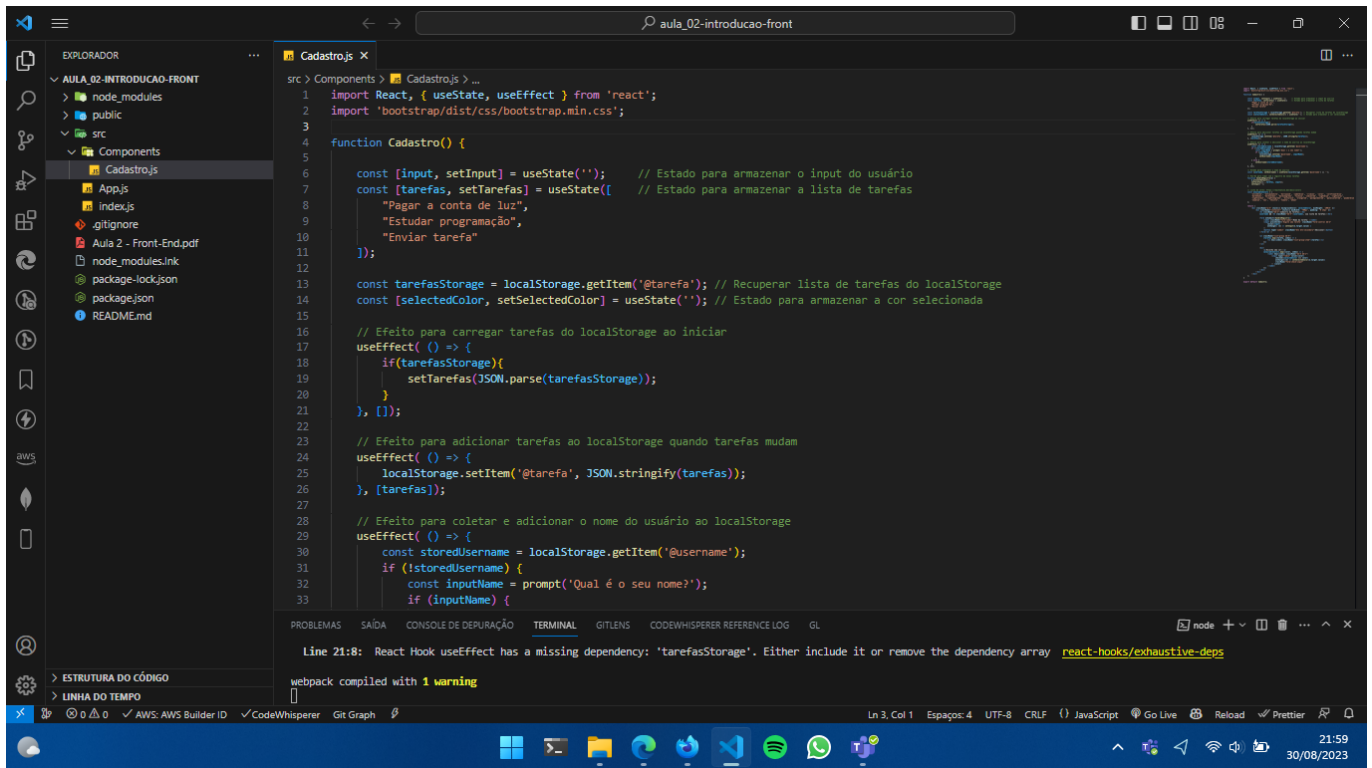
- A. Utilizando Hooks e localStorage, ao usuário entrar na página um alert deve aparecer perguntando ao usuário seu nome.
- B. Ao usuário informar seu nome o valor deve ser salvo no localStorage
- C. Usando Hooks, faça a leitura do valor salvo no localStorage e insira na tela o nome do usuário e depois a frase “, sua lista de tarefas”
- D. Crie alguns Radio Buttons com nome de cores. Utilize a documentação do React: <https://reactnative.dev/docs/colors>
- E. Ao usuário escolher uma cor o fundo da página deverá receber a mesma cor

Solução:









```

1 import React, { useState, useEffect } from 'react';
2 import 'bootstrap/dist/css/bootstrap.min.css';
3
4 function Cadastro() {
5
6     const [input, setInput] = useState(''); // Estado para armazenar o input do usuário
7     const [tarefas, setTarefas] = useState([ // Estado para armazenar a lista de tarefas
8         "Pagar a conta de luz", "Estudar programação", "Enviar tarefa"
9     ]);
10    const tarefasStorage = localStorage.getItem('@tarefas'); // Recuperar lista de tarefas do localStorage
11    const [selectedColor, setSelectedColor] = useState(''); // Estado para armazenar a cor selecionada
12    const [username, setUsername] = useState(localStorage.getItem('@username') || ''); // Estado para armazenar o nome do usuário
13    const availableColors = [ // Lista de cores: https://reactnative.dev/docs/colors
14        'aliceblue', 'antiquewhite', 'burlywood', 'cadetblue', 'crimson', 'coral', 'cornflowerblue',
15        'darkkhaki', 'darksalmon', 'darkseagreen', 'fuchsia', 'indianred', 'lightblue', 'lightcoral',
16        'lightsalmon', 'lightseagreen', 'mistyrose', 'orangered', 'palegoldenrod', 'palevioletred', 'powderblue',
17        'skyblue', 'tan', 'thistle', 'tomato', 'wheat'
18    ];
19
20    useEffect( () => { // Efeito para carregar tarefas do localStorage ao iniciar
21        if(tarefasStorage){
22            setTarefas(JSON.parse(tarefasStorage));
23        }
24    }, []);
25
26    useEffect( () => { // Efeito para adicionar tarefas ao localStorage quando tarefas mudam
27        localStorage.setItem('@tarefas', JSON.stringify(tarefas));
28    }, [tarefas]);
29
30    useEffect( () => { // Efeito para coletar e adicionar o nome do usuário ao localStorage
31        const storedUsername = localStorage.getItem('@username');
32        if (!storedUsername) {
33            const inputName = prompt('Qual é o seu nome?');
34            if (inputName) {
35                localStorage.setItem('@username', inputName);
36                setUsername(inputName);
37            }
38        } else {
39            setUsername(storedUsername);
40        }
41    }, []);
42
43    function handleRegistro(e) { // Função para lidar com o registro de novas tarefas
44        e.preventDefault();
45        setTarefas([...tarefas, input]);
46        setInput('');
47    }
48
49    return (
50        <div className="p-4" style={{ backgroundColor: selectedColor, minHeight: '100vh' }}>
51            <div className="container" style={{ maxWidth: '700px', padding: '0 15px' }}>
52                <h1 className="mb-2">Cadastro de Tarefas:</h1>
53                <h2 className="mb-3">{username}, sua lista de tarefas:</h2>
54                <form onSubmit={handleRegistro}>
55                    <label className="form-label">Nome da tarefa: </label>
56                    <input placeholder="Digite uma tarefa" className="form-control mb-2"
57                        value={input}
58                        onChange={ (e) => setInput(e.target.value) }/>
59                    <button type="submit" className="btn btn-secondary">Adicionar</button>
60                </form><br />
61                <ul className="list-group mb-4">
62                    {tarefas.map((tarefa, index) => (
63                        <li key={index} className="list-group-item">{tarefa}</li>
64                    ))}
65                </ul>
66                <div>
67                    <p>Escolha uma cor:</p>
68                    {availableColors.map((color, index) => (
69                        <label key={index} className="me-3 mb-1">
70                            <input type="radio" value={color}
71                                checked={selectedColor === color}
72                                onChange={(e) => setSelectedColor(e.target.value)}
73                                className="form-check-input" /> {color}
74                        </label>
75                    ))}
76                </div>
77            </div>
78        </div>
79    );
80 }
81
82 export default Cadastro;

```