

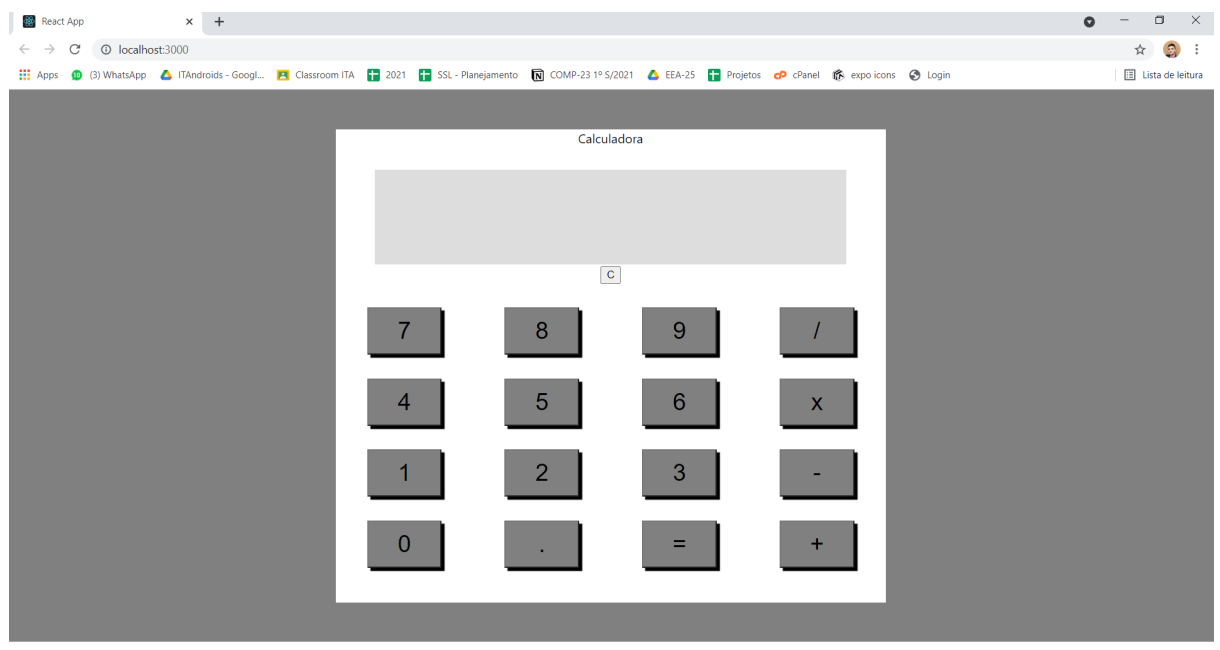


Instituto Tecnológico de Aeronáutica
Divisão de Ciência da Computação - IEC
CES-26 - COMP 23
6ª série de exercícios

Arthur José de Sousa Rodrigues

Ferramentas principais utilizadas: React Js, Redux

1) Front-end da calculadora:

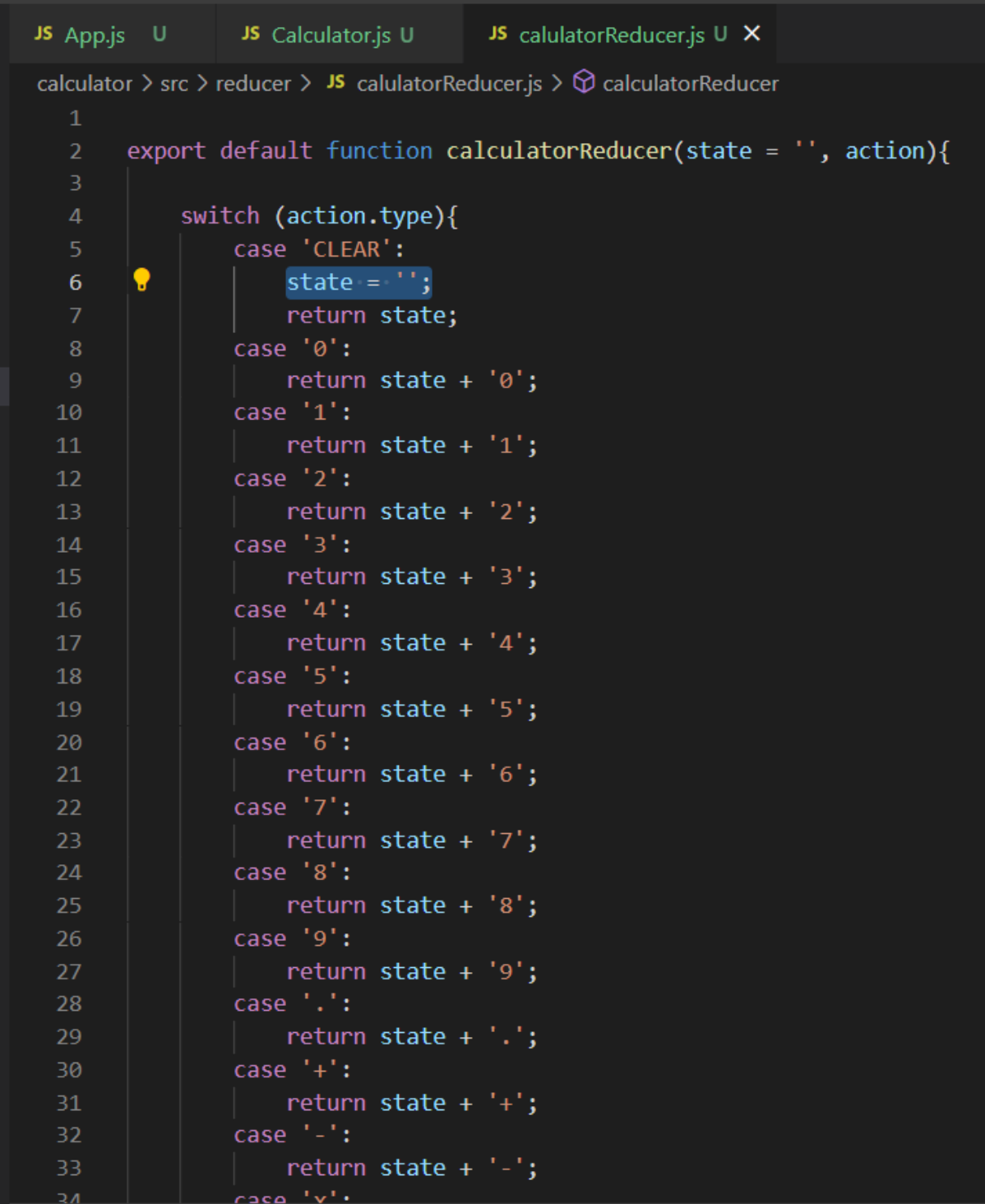


Para se criar o front-end da calculadora, criou-se um componente chamado `calculator`, conforme consta em `components/Calculator.js`.

- 2) Utilizou-se os módulos *redux* e *react-redux* para que se pudesse utilizar o gerenciamento de estados no framework do react. O componente *Provider* (do módulo *react-redux*) foi utilizado em *App.js* para que se pudesse gerenciar os estados por todo o app. Além disso criou-se uma pasta chamada *reducer*, que contém o script *calculatorReducer.js*, o qual possui a implementação de fato do gerenciamento de estados.

```
JS App.js U X JS Calculator.js U JS calculatorReducer.js U
calculator > src > JS App.js > App
1  import './App.css';
2  import Calculator from './components/Calculator';
3  import {createStore} from 'redux';
4  import calculatorReducer from './reducer/calculatorReducer';
5  import {Provider} from 'react-redux';
6
7  function App() {
8
9      const store = createStore(calculatorReducer);
10
11     return (
12         <div className="App">
13             <Provider store={store}>
14                 <Calculator></Calculator>
15             </Provider>
16         </div>
17     );
18 }
19
20 export default App;
21
```

- 3) Em `reducer/calculatorReducer.js`, está toda a implementação da lógica da calculadora, conforme mostra as Figuras abaixo. Vale ressaltar que a calculadora está simples e não possui um tratamento de erros avançado.



```
calculator > src > reducer > JS calculatorReducer.js > calculatorReducer
1
2 export default function calculatorReducer(state = '', action){
3
4     switch (action.type){
5         case 'CLEAR':
6             state = '';
7             return state;
8         case '0':
9             return state + '0';
10        case '1':
11            return state + '1';
12        case '2':
13            return state + '2';
14        case '3':
15            return state + '3';
16        case '4':
17            return state + '4';
18        case '5':
19            return state + '5';
20        case '6':
21            return state + '6';
22        case '7':
23            return state + '7';
24        case '8':
25            return state + '8';
26        case '9':
27            return state + '9';
28        case '.':
29            return state + '.';
30        case '+':
31            return state + '+';
32        case '-':
33            return state + '-';
34        case 'x':
```

```
    return state + 'x';
case '/':
    return state + '/';
case '=':
    if(state.indexOf('+') != -1){
        let numbers = state.split('+');
        if(numbers[0].indexOf('.') != -1 || numbers[1].indexOf('.') != -1){
            state = (parseFloat(numbers[0]) + parseFloat(numbers[1])).toString();
        }
        else{
            state = (parseInt(numbers[0]) + parseInt(numbers[1])).toString();
        }
        return state;
    }
    else if(state.indexOf('-') != -1){
        let numbers = state.split('-');
        if(numbers[0].indexOf('.') != -1 || numbers[1].indexOf('.') != -1){
            state = (parseFloat(numbers[0]) - parseFloat(numbers[1])).toString();
        }
        else{
            state = (parseInt(numbers[0]) - parseInt(numbers[1])).toString();
        }
        return state;
    }
    else if(state.indexOf('x') != -1){
        let numbers = state.split('x');
        if(numbers[0].indexOf('.') != -1 || numbers[1].indexOf('.') != -1){
            state = (parseFloat(numbers[0]) * parseFloat(numbers[1])).toString();
        }
        else{
            state = (parseInt(numbers[0]) * parseInt(numbers[1])).toString();
        }
        return state;
    }
}
```

```

    }
    else if(state.indexOf('/') !== -1){
      let numbers = state.split('/');
      if(numbers[0].indexOf('.') !== -1 || numbers[1].indexOf('.') !== -1){
        state = (parseFloat(numbers[0]) / parseFloat(numbers[1])).toString();
      }
      else{
        state = (parseInt(numbers[0]) / parseInt(numbers[1])).toString();
      }
      return state;
    }
    else{
      return state;
    }
  }
  default: return state;
}
}

```

- 4) No componente *Calculator.js*, a função *calculatorReducer* é chamada através do método *useDispatch()* do módulo *react-redux*. Assim, quando o usuário clica em um botão, a requisição de mudança de estado do display é despachada pelo *redux*.

```

JS App.js U    JS Calculator.js U X    JS calculatorReducer.js U
calculator > src > components > JS Calculator.js > Calculator
1  import React from 'react';
2  import {useSelector, useDispatch} from 'react-redux';
3
4  function Calculator(props) {
5
6    const display = useSelector((state) => {return state});
7
8    const dispatch = useDispatch();
9
10   return (
11     <div className="main" style={styles.main}>
12       <div className="calculatorContainer" style={styles.calculatorContainer}>
13         Calculadora
14         <div className="header" style={styles.header} id="display">
15           {display}
16         </div>
17         <button className="bodyRowElement" style={styles.clearbutton} onClick={() => {dispatch({type: 'CLEAR'})}}>
18           C
19         </button>
20         <div className="bodyContainer">
21           <div className="bodyRowContainer" style={styles.bodyRowContainer}>
22             <button className="bodyRowElement" style={styles.bodyRowElement} onClick={() => {dispatch({type: '7'})}}>
23               7
24             </button>
25             <button className="bodyRowElement" style={styles.bodyRowElement} onClick={() => {dispatch({type: '8'})}}>
26               8
27             </button>
28             <button className="bodyRowElement" style={styles.bodyRowElement} onClick={() => {dispatch({type: '9'})}}>
29               9
30             </button>
31             <button className="bodyRowElement" style={styles.bodyRowElement} onClick={() => {dispatch({type: '/'})}}>
32               /
33           </div>

```

EXEMPLOS

- 1.2 + 10.8:

Calculadora

1.2+10.8

C

7

8

9

/

4

5

6

x

1

2

3

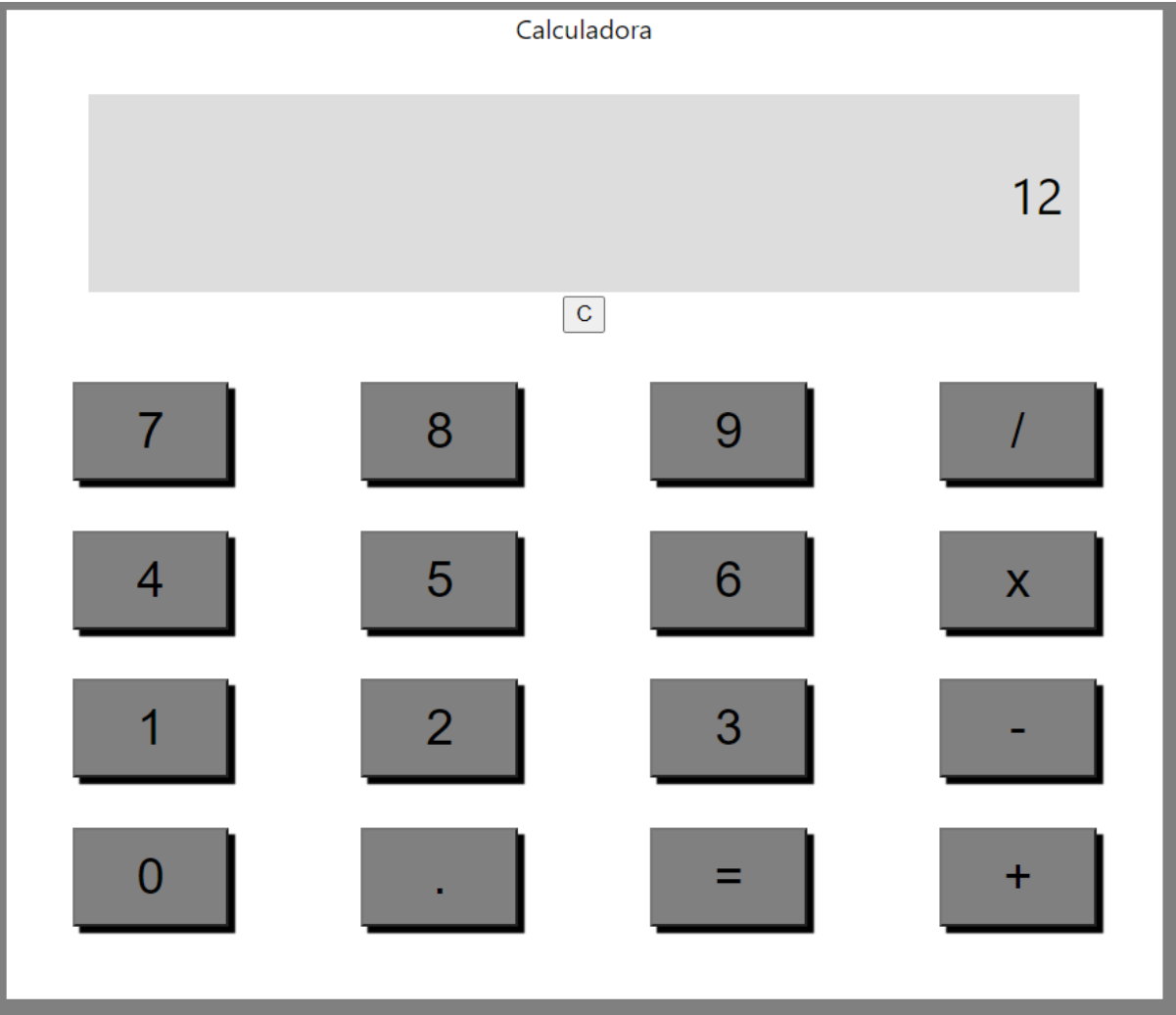
-

0

.

=

+



- 32 x 45:

Calculadora

32x45

C

7

8

9

/

4

5

6

x

1

2

3

-

0

.

=

+

Calculadora

1440

C

7

8

9

/

4

5

6

x

1

2

3

-

0

.

=

+