

```
import Input from './components/Input';
```

```
import Button from './components/Button';
```

```
import { Container, Content, Row } from './styles';
```

```
import { useState } from 'react';
```

```
const App = () => {
```

```
  const [currentNumber, setCurrentNumber] = useState('0');
```

```
  const [firstNumber, setFirstNumber] = useState('0');
```

```
  const [operation, setOperation] = useState("");
```

```
  const handleOnClear = () => {
```

```
    setCurrentNumber('0');
```

```
    setFirstNumber('0');
```

```
    setOperation("");
```

```
  };
```

```
  // Função para deletar um número por vez
```

```
  const handleBack = () => {
```

```
    setCurrentNumber(prev => (prev.length > 1 ? prev.slice(0, -1) : '0'));
```

```
  };
```

```
  const handleAddNumber = (num) => {
```

```
    setCurrentNumber(prev => `${prev === '0' ? "" : prev}${num}`);
```

```
  };
```

```
  const handleSumNumbers = () => {
```

```
    if (firstNumber === '0') {
```

```
      setFirstNumber(String(currentNumber));
```

```
    setCurrentNumber('0');  
    setOperation('+');  
  } else {  
    const sum = Number(firstNumber) + Number(currentNumber);  
    setCurrentNumber(String(sum));  
    setOperation("");  
  }  
};
```

```
const handleMinusNumbers = () => {  
  if (firstNumber === '0') {  
    setFirstNumber(String(currentNumber));  
    setCurrentNumber('0');  
    setOperation('-');  
  } else {  
    const difference = Number(firstNumber) - Number(currentNumber);  
    setCurrentNumber(String(difference));  
    setOperation("");  
  }  
};
```

// Função para multiplicação

```
const handleMultiplyNumbers = () => {  
  if (firstNumber === '0') {  
    setFirstNumber(String(currentNumber));  
    setCurrentNumber('0');  
    setOperation('*');  
  } else {  
    const product = Number(firstNumber) * Number(currentNumber);
```

```
    setCurrentNumber(String(product));  
    setOperation("");  
  }  
};
```

// Função para divisão

```
const handleDivideNumbers = () => {  
  if (firstNumber === '0') {  
    setFirstNumber(String(currentNumber));  
    setCurrentNumber('0');  
    setOperation('/');  
  } else {  
    const quotient = Number(firstNumber) / Number(currentNumber);  
    setCurrentNumber(String(quotient));  
    setOperation("");  
  }  
};
```

// Função para Pi

```
const handlePi = () => {  
  setCurrentNumber(String(Math.PI));  
};
```

// Função para raiz quadrada

```
const handleSquareRoot = () => {  
  setCurrentNumber(String(Math.sqrt(Number(currentNumber))));  
};
```

```
const handleEquals = () => {
```

```

if (firstNumber !== '0' && operation !== " " && currentNumber !== '0') {
  switch (operation) {
    case '+':
      handleSumNumbers();
      break;
    case '-':
      handleMinusNumbers();
      break;
    case '*':
      handleMultiplyNumbers();
      break;
    case '/':
      handleDivideNumbers();
      break;
    default:
      break;
  }
}
};

```

```

return (
  <Container>
    <Content>
      <Input value={currentNumber} />
      <Row>
        <Button label="√" onClick={handleSquareRoot} />
        <Button label="π" onClick={handlePi} />
        <Button label="C" onClick={handleOnClear} />
        <Button label="÷" onClick={handleDivideNumbers} />
      </Row>
    </Content>
  </Container>
);

```

```

</Row>

<Row>

  <Button label="7" onClick={() => handleAddNumber('7')} />
  <Button label="8" onClick={() => handleAddNumber('8')} />
  <Button label="9" onClick={() => handleAddNumber('9')} />
  <Button label="x" onClick={handleMultiplyNumbers} />

</Row>

<Row>

  <Button label="4" onClick={() => handleAddNumber('4')} />
  <Button label="5" onClick={() => handleAddNumber('5')} />
  <Button label="6" onClick={() => handleAddNumber('6')} />
  <Button label="-" onClick={handleMinusNumbers} />

</Row>

<Row>

  <Button label="1" onClick={() => handleAddNumber('1')} />
  <Button label="2" onClick={() => handleAddNumber('2')} />
  <Button label="3" onClick={() => handleAddNumber('3')} />
  <Button label="+" onClick={handleSumNumbers} />

</Row>

<Row>

  <Button label="<" onClick={handleBack} />
  <Button label="0" onClick={() => handleAddNumber('0')} />
  <Button label="," onClick={() => handleAddNumber(',')} />
  <Button label="=" onClick={handleEquals} />

</Row>

</Content>

</Container>

);
}

```

```
export default App;
```