

Lista de Exercício de Apps Gabriel Batalha Michelini
Professor: Victor Ferrari

1)a)

```
import { useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'

function OlaMundo() {
  return (

    <h1>Bem vindo ao React!</h1>

  );
}

export default OlaMundo
```

b)

```
import { useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'

function OlaMundo() {
  return (

    <h1>Olá mundo!</h1>

  );
}

export default OlaMundo
```

c)

```
import { useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'

function Olahora() {
  const agora = new Date();
```

```

    const diasdasemana = ["domingo", "segunda-feira", "terça-feira",
"quarta-feira", "quinta-feira", "sexta-feira", "sábado"];
    const meses = ["janeiro", "fevereiro", "março", "abril", "maio",
"junho", "julho", "agosto", "setembro", "outubro", "novembro",
"dezembro"];
    const diaSemana = diasdasemana[agora.getDay()];
    const dia = agora.getDate();
    const mes = meses[agora.getMonth()];
    const ano = agora.getFullYear();
    return (

        <h1>Hoje é {diaSemana}, {dia} de {mes} de {ano}</h1>

    );
}

export default Olahora

```

2)a)

```

import { useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'

function Saudacao({nome}) {
    return (
        <h1>Olá , {nome}!</h1>
    );
}

export default function Myapp(){
    return (<Saudacao nome = "Gabriel"/>)
}

```

b)

```

import { useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'

```

```
function Saudacao({usuario}) {
  return (
    <h1>Boas vindas , {usuario}!</h1>
  );
}

export default function MyApp(){
  return (<Saudacao usuario = "Gabriel"/>)
}
```

c)

```
function StatusJogo({nivel, pontos}) {
  const maxPontos = nivel * 1000;
  const progresso = Math.min((pontos/maxPontos) * 100, 100);
  const corBarra = progresso < 30 ? "red" : progresso < 70 ? "orange" :
"green";

  return (
    <div style={{padding: "20px"}}>
      <h2>Nível {nivel}</h2>
      <p>Pontos: {pontos} / {maxPontos}</p>
      <div style={{
        width: "200px",
        height: "20px",
        backgroundColor: "#ddd",
        borderRadius: "10px",
        overflow: "hidden"
      }}>
        <div style={{
          width: `${progresso}%`,
          height: "100%",
          backgroundColor: corBarra,
          transition: "all 0.3 ease",
        }}></div>
      </div>
      <p>{progresso.toFixed(1)}% completo</p>
    </div>
  );
}
```

```
export default function Myapp() {  
  return <StatusJogo nivel={5} pontos={2500}/>  
}
```

```
3)a)function ListaHobbies() {  
  const hobbies = ["Leitura", "Natação", "programação", "Culinária"]  
  
  return (  
    <div>  
      <h2>Meus Hobbies Favoritos</h2>  
      <ul>  
        {hobbies.map((hobby, index) => (<li key={index}> {hobby}</li>  
          ))}  
      </ul>  
    </div>  
  );  
}  
  
export default ListaHobbies
```

```
b)import { useState } from 'react'  
import reactLogo from './assets/react.svg'  
import viteLogo from '/vite.svg'  
import './App.css'  
  
function ListaComida() {  
  const comidas = ["Lasanha", "Bife de chorizo", "Pizza", "Macarrão a  
carbonara"]  
  
  return (  
    <div>  
      <h2>Minhas comidas favoritas</h2>  
      <ul>  
        {comidas.map((comida, index) => (<li key={index}> {comida}</li>  
          ))}  
      </ul>  
    </div>  
  );  
}
```

```
export default ListaComida
```

```
.prato-card{
  color: rgb(255, 0, 55);
  border: 4px solid black;
  padding: 0 0 10px 0;
  border-radius: 10px;
  margin: 20px;
  max-width: 10;
}

.menu-grid{
  color: black;
}
```

c)function MenuRestaurante() {

```
  const pratos = [
    {nome: "Bife de Chorizo", preco: 50.00, descricao: "Um bife de chorizo ao ponto. Feito pelo chefe Michelin com suas 5 estrelas Michelin"},
    {nome: "Macarrão a Carbonara", preco: 39.90, descricao: "Um macarrão cremoso com molho branco e bacon picado. Feito pelo chefe Michelin"},
    {nome: "Pizza De Calabresa", preco: 40.00, descricao: "Pizza de 8 pedaços com queijo e um molho de tomate artesanal além de calabresa pi"},
    {nome: "Palha Italiana Gourmet", preco: 20.90, descricao: "Uma sobremesa que consiste em um brigadeiro com biscoito gourmet. Feito pelo c"}
  ];

  return (
    <div>
      <h1>Cardápio do Restaurante Michelin</h1>
      <div className='menu-grid'>
        {pratos.map((prato, index) => (
          <div key={index} className='prato-card'>
            <h3>{prato.nome}</h3>
            <p className='preco'>R${prato.preco.toFixed(2)}</p>
            <p className='descricao'>{prato.descricao}</p>
          </div>
        ))}
      </div>
    </div>
  )
}
```

```

    </div>
  </div>
);
}

```

4)a) `function Cartaopessoa({nome, idade, profissao}) {`

```

  return (
    <div className='cartao-pessoa'>
      <h2>{nome}</h2>
      <p><strong>Idade:</strong> {idade} anos</p>
      <p><strong>Profissao:</strong> {profissao}</p>
    </div>
  );
}

function App() {
  return(
    <div>
      <Cartaopessoa nome = "Gabriel Michelinini" idade={18} profissao="
desenvolvedor" />
    </div>
  );
}

export default App

```

b)

```

function CartaoLivro({titulo, autor, genero}) {
  return (
    <div className='cartao pessoa'>
      <h2>{titulo}</h2>
      <p><strong>autor:</strong> {autor}</p>
      <p><strong>Genero:</strong> {genero}</p>
    </div>
  );
}

function App() {
  return(
    <div>

```

```

        <Cartaolivro titulo="As crônicas de michel" autor="Gabriel
Michellini" genero="Romance" />
      </div>
    );
  }

export default App

```

c)

```

function PrevisaoTempo({temperatura, clima, cidade, umidade}) {
  const getIcone = (clima) => {
    const climas = {
      "ensolarado": "☀️",
      "nublado": "☁️",
      "chuvoso": "🌧️",
      "tempestuoso": "⚡️",
      "nevando": "❄️",
    };
    return climas[clima.toLowerCase()] || "🌡️";
  };

  const getCorTemperatura = (temp) => {
    if (temp < 15) return "#00bfff";
    if (temp < 25) return "#ffa500";
    return "#d0021b";
  };

  return (

```



```

        <PrevisaoTempo Temperatura={-2} clima="nevando" cidade="Rio de
Janeiro" umidade={60} />
        <PrevisaoTempo Temperatura={24} clima="nublado" cidade="sao
paulo" umidade={70} />
        <PrevisaoTempo Temperatura={100} clima="ensolarado" cidade="Iraja
umidade={70}" />
    </div>
  );
}

export default App;

```

5)a) `import { useState } from 'react';`

```

function Contador() {
  const [count, setCount] = useState(0);
  return (
    <div>
      <h2>Contador: {count}</h2>
      <button onClick={() => setCount(count + 1)}>+1</button>
      <button onClick={() => setCount(count - 1)}>-1</button>
    </div>
  );
}

export default Contador;

```

b) `import './App.css'`

```

import { useState } from 'react';

function Contador() {
  const [count, setCount] = useState(10);
  return(
    <div>
      <h2> Contador: {count}</h2>
      <button onClick={() => setCount(count + 5)}>+5</button>
      <button onClick={() => setCount(count - 5)}>-5</button>
    </div>
  )
}

export default Contador

```

```

c)import { useState } from 'react';
import './App.css';
function Termometro() {
  const [count, setCount] = useState(20);
  const getCorTemperatura = (count) => {
    if (count < 20) return "#4a90e2";
    if (count > 21) return "#ee2828ff";
  }

  return(
    <div>
      <h1>Termômetro Digital</h1>
    </div>

    <div style={{color:getCorTemperatura(count)}}>
      <h2 className="bonito1"> Temperatura: {count} °C</h2>
      <button className="botao2" onClick={() => setCount(count + 2)}>🔥
Aquecer(+2 °C)</button>
      <button className="botao1" onClick={() => setCount(count - 2)}>❄️
Esfriar(-2 °C)</button>
    </div>
  )
}

export default Termometro

```

```

.botao2{
  background-color: red;
}
.botao1{
  background-color: blue;
}
.bonito1{
  background-color: yellow;
}

```

6)a)import { useState } from 'react'; // Adicionado pois `useState` é utilizado

```

function MostrarEsconder() {
  const [mostrar, setMostrar] = useState(true);
  return(

```

```

    <div>
      <button onClick={() => setMostrar(!mostrar)}>
        {mostrar ? 'Esconder' : 'Mostrar'}
      </button>
      {mostrar && <p>Presta atenção na aula</p>}
    </div>
  )
}

export default MostrarEsconder

```

b)

```

import { useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'

function MostrarEsconder() {
  const [mostrar, setMostrar] = useState(true);
  return(
    <div>
      <button onClick={() => setMostrar(!mostrar)}>
        {mostrar ? 'Esconder' : 'Mostrar'}
      </button>
      {mostrar && <img src='src/assets/images.png' alt="trailer"/>}
    </div>
  )
}

export default MostrarEsconder

```

c)

```

import React, { useState } from 'react';

```

```

function Lista_Itens() {
  const itens = [
    "computador",
    "celular",
    "balas",
    "chocolates",
    "album da copa 2026"
  ];
}

```

```

    return (
      <div>
        <ul>
          {itens.map((item, index) => (
            <li key={index}>{item}</li>
          ))}
        </ul>
      </div>
    );
  }

function AbrirFechar() {
  const [abrir, setAbrir] = useState(true);

  return (
    <div>
      <div className='cofre'>
        <p>Cofre Digital</p>
      </div>
      <div>
        <p>🔒</p>
      </div>
      <div>
        <button onClick={() => setAbrir(!abrir)}>
          {abrir ? 'fechar 🔒' : 'abrir 🔒'}
        </button>

        {abrir && <Lista_Itens />}

      </div>
    </div>
  );
}

export default AbrirFechar;

```

7)a) `import React, { useState } from 'react';`

```

function InputTempoReal() {
  const [texto, setTexto] = useState('');

  return (

```

```

    <div>
      <input
        type="text"
        value={texto}
        onChange={ (e) => setTexto(e.target.value) }
        placeholder='digite algo:'
      />
      <p>Você digitou: {texto}</p>
    </div>
  );
}

export default InputTempoReal;

```

b) `import React, { useState } from 'react';`

```

function InputTempoReal() {
  const [texto, setTexto] = useState('');
  const palavraM = texto.toUpperCase();

  return (
    <div>
      <input
        type="text"
        value={texto}
        onChange={ (e) => setTexto(e.target.value) }
        placeholder='digite algo:'
      />
      <h2>Você digitou: {palavraM}</h2>
    </div>
  );
}

export default InputTempoReal;

```

c) `import React, { useState } from 'react';`

```

function InputTempoReal() {
  const [texto, setTexto] = useState('');
  const textom = texto.length;

```

```

return (
  <div>
    <input
      type="text"
      value={texto}
      onChange={ (e) => setTexto(e.target.value)}
      placeholder='digite algo:'
    />

    <h1>Você digitou: {texto}</h1>

    { /* Renderização Condicional */ }
    {textom > 0 && textom < 3 && (
      <h2>Menos de 3 caracteres</h2>
    )}

    {textom > 3 && (
      <h3>Mais de 3 caracteres</h3>
    )}
  </div>
);
}

export default InputTempoReal;

```

```

h2 {
  color: red;
}

h3 {
  color: green;
}

```

d) `import React, { useState } from 'react';`

```

function InputTempoReal() {
  const [texto, setTexto] = useState('');
  const textom = texto.length;

  return (
    <div>
      <input
        type="text"

```

```

        maxLength={50}
        value={texto}
        onChange={(e) => setTexto(e.target.value)}
        placeholder='digite algo:'
      />

      <h1>Você digitou: {texto}</h1>
      <h1>Total de caracteres {textom}/50</h1>

      {textom > 0 && textom < 3 && (
        <h2>Menos de 3 caracteres</h2>
      )}

      {textom >= 3 && (
        <h3>Mais de 3 caracteres</h3>
      )}
    </div>
  );
}

export default InputTempoReal;

```

e)export default BotaoCurtir;

```

import React, { useState } from 'react';

function GeradorDesenha() {
  const [texto, setTexto] = useState('');
  const textom = texto.length;
  const textoI = texto.split('').reverse().join('');
  const caracteresE = "*&%";

  return (
    <div>
      <h1>Criptografador de senhas!!</h1>
      <p>Digite uma palavra para ser criptografada: {texto}</p>
      <input
        type="text"
        maxLength={30}
        value={texto}
        onChange={(e) => setTexto(e.target.value)}
        placeholder='digite algo:'
      />
    </div>
  );
}

```

```

    {textom > 0 && textom < 30 && (
      <h3>
        Sua senha criptografada é {textoI}
        {textom}
        {textom}
        {caracteresE}
      </h3>
    )}

    <div className='a'>
      {textom > 0 && textom < 6 && (
        <h4>Sua senha é fraca</h4>
      )}
    </div>

    <div className='b'>
      {textom > 0 && textom >= 6 && textom <= 10 && (
        <h4>Sua senha é media</h4>
      )}
    </div>

    <div className='c'>
      {textom > 0 && textom > 10 && textom < 30 && (
        <h4>Sua senha é forte</h4>
      )}
    </div>

    {textom > 0 && textom > 30 && (
      <h4>Sua senha é muito grande </h4>
    )}
  </div>
);
}

export default GeradorDesenha;

```


Criptografador de senhas!!

Digite uma palavra para ser criptografada: safadoooooooooooo

safadoooooooooooo

Sua senha criptografada é oooooooooooooodafas1818*&%

Sua senha é forte

8)a) `import React, { useState } from 'react';`

```
function AdicionarLista() {
  const [item, setItem] = useState('');
  const [lista, setLista] = useState([]);

  const adicionarItem = () => {
    if (item.trim()) {
      setLista([...lista, item]);
      setItem("");
    }
  };

  return (
    <div>
      <input
        type="text"
        value={item}
        onChange={(e) => setItem(e.target.value)}
        placeholder='Digite um Item'
      />
      <button onClick={adicionarItem}>Adicionar</button>
      <ul>
        {lista.map((itemLista, index) => (
          <li key={index}>{itemLista}</li>
        ))}
      </ul>
    </div>
  );
}
```

```

        </ul>
      </div>
    );
  }

export default AdicionarLista;

```

b) `import React, { useState } from 'react';`

```

function AdicionarLista() {
  const [item, setItem] = useState('');
  const [lista, setLista] = useState([]);

  const adicionarItem = () => {
    if (item.trim()) {
      setLista([...lista, item]);
      setItem("");
    }
  };

  return (
    <div>
      <input
        type="text"
        value={item}
        onChange={e => setItem(e.target.value)}
        placeholder='Digite um Item'
      />
      <button onClick={adicionarItem}>Adicionar</button>
      <ul>
        {lista.map((itemLista, index) => (
          <li key={index}>{itemLista}</li>
        ))}
      </ul>
    </div>
  );
}

export default AdicionarLista;

```

c)

```

import React, { useState } from 'react';

```

```

function AdicionarLista() {
  const [item, setItem] = useState('');
  const [lista, setLista] = useState([]);
  const [pontos, setPontos] = useState("");

  const adicionarTime = () => {
    if (item.trim() && pontos.trim() && !isNaN(pontos)) {
      const novoTime = {
        nome: item.trim(),
        pontos: parseInt(pontos, 10),
      };
      const novaLista = [...lista, novoTime];
      novaLista.sort((a, b) => b.pontos - a.pontos);
      setLista(novaLista);
      setItem("");
      setPontos("");
    } else {
      alert("Por favor, preencha o nome do time e uma pontuação válida.");
    }
  };

  return (
    <div>
      <h1>Tabela de Pontos</h1>
      <input
        type="text"
        value={item}
        onChange={(e) => setItem(e.target.value)}
        placeholder='Digite um time'
      />
      <input
        type="number"
        value={pontos}
        onChange={(e) => setPontos(e.target.value)}
        placeholder='Pontuação'
      />
      <button onClick={adicionarTime}>Adicionar</button>
      <ul>
        {lista.map((time, index) => (
          <li key={index}>{time.nome}-{time.pontos}</li>
        ))}
      </ul>
    </div>
  );
}

```

```

    </div>

  );
}

export default AdicionarLista;

```

d)

9)a)

```

import { useState } from 'react';
import './App.css';

function AlterarcorFundo() {
  const [cor, setCor] = useState("white");

  const mudarCor = (novaCor) => {
    setCor(novaCor);
    document.body.style.backgroundColor = novaCor;
  };

  return (
    <div>
      <h2>Alterar a cor de fundo</h2>
      <button onClick={() => mudarCor("lightblue")}>Azul</button>
      <button onClick={() => mudarCor("lightgreen")}>Verde</button>
      <button onClick={() => mudarCor("lightcoral")}>Coral</button>
      <button onClick={() => mudarCor("white")}>Branco</button>
    </div>
  );
}

export default AlterarcorFundo;

```

b)

```

import { useState } from 'react';
import './App.css';

function AlterarcorFundo() {
  const [tam, setTam] = useState("100px");
  const mudarTam = (novaTam) => {
    setTam(novaTam);
    document.body.style.fontSize=novaTam;
  };
  return (
    <div>

```

```

        <h2 > Alterar o Tamanho da Fonte</h2>
        <button onClick={() => mudarTam("50px")}>Pequeno</button>
        <button onClick={() => mudarTam("100px")}>Medio</button>
        <button onClick={() => mudarTam("200px")}>Grande</button>

      </div>
    );
  }

```

```
export default AlterarcorFundo;
```

```

c)import { useState } from 'react';
import './App.css';
const humores = {
  feliz: {
    corFundo: 'lightyellow',
    emoji: '😄',
    fala: 'Haha!',
    corTexto: 'black'
  },
  triste: {
    corFundo: 'blue',
    emoji: '😞',
    fala: 'Sniff...',
    corTexto: 'white'
  },
  raivoso: {
    corFundo: 'red',
    emoji: '😡',
    fala: 'Grrrr!',
    corTexto: 'white'
  },
  calmo: {
    corFundo: 'white',
    emoji: '😌',
    fala: 'Ahhhh...',
    corTexto: 'black'
  }
};

function SimuladordeHumor() {
  const [humor, setHumor] = useState(humores.calmo);
  const mudarHumor = (nomeHumor) => {
    const novoHumor = humores[nomeHumor];

```

```

    setHumor(novoHumor);

    document.body.style.backgroundColor = novoHumor.corFundo;
    document.body.style.color = novoHumor.corTexto;
    alert(novoHumor.fala);
  };

  return (
    <div >
      <h1>Como você está se sentindo hoje?</h1>
      <div className="emoji-display">
        {humor.emoji}
      </div>
      <div>
        <button onClick={() => mudarHumor("feliz")}>Feliz</button>
        <button onClick={() => mudarHumor("triste")}>Triste</button>
        <button onClick={() => mudarHumor("raivoso")}>Raivoso</button>
        <button onClick={() => mudarHumor("calmo")}>Calmo</button>
      </div>
    </div>
  );
}

export default SimuladordeHumor;

```

10)a)

```

import { useState } from 'react';
import './App.css';

function Calculadora() {
  const [num1, setNum1] = useState('');
  const [num2, setNum2] = useState('');
  const [resultado, setResultado] = useState(null);

  const handleOperacao = (operador) => {
    const n1 = parseFloat(num1);
    const n2 = parseFloat(num2);

    if (isNaN(n1) || isNaN(n2)) {
      setResultado('Erro: Insira números válidos');
      return;
    }

    let res;
    switch (operador) {

```

```
    case '+':
      res = n1 + n2;
      break;
    case '-':
      res = n1 - n2;
      break;
    case '*':
      res = n1 * n2;
      break;
    case '/':
      if (n2 === 0) {
        setResultado('Erro: Divisão por zero');
        return;
      }
      res = n1 / n2;
      break;
    default:
      return;
  }

  setResultado(res);
};

const limpar = () => {
  setNum1('');
  setNum2('');
  setResultado(null);
};

return (
  <div className="calculadora-container">
    <h1>Calculadora Simples</h1>

    <div className="inputs-container">
      <input
        type="number"
        value={num1}
        onChange={(e) => setNum1(e.target.value)}
        placeholder="Primeiro número"
      />
      <input
        type="number"
        value={num2}
```

```

        onChange={ (e) => setNum2(e.target.value) }
        placeholder="Segundo número"
      />
    </div>

    <div className="botoes-container">
      <button onClick={ () => handleOperacao('+') }>+</button>
      <button onClick={ () => handleOperacao('-') }>-</button>
      <button onClick={ () => handleOperacao('*') }>*</button>
      <button onClick={ () => handleOperacao('/') }>/</button>
      <button onClick={limpar}
className="botao-limpar">Limpar</button>
    </div>

    {resultado !== null && (
      <h3 className="resultado">
        Resultado: {resultado}
      </h3>
    )}
  </div>
);
}

export default Calculadora;

```

```

b)import { useState } from 'react';
import './App.css';

```

```

function Calculadora() {
  const [num1, setNum1] = useState('');
  const [num2, setNum2] = useState('');
  const [resultado, setResultado] = useState(null);

  const handleOperacao = (operador) => {
    const n1 = parseFloat(num1);
    const n2 = parseFloat(num2);

    if (isNaN(n1) || isNaN(n2)) {
      setResultado('Erro: Insira números válidos');
      return;
    }

    let res;
    switch (operador) {

```



```

    case 'Calcular a area':
      res = n1 * n2;
      break;
    case '/':
      if (n2 === 0) {
        setResultado('Erro: Divisão por zero');
        return;
      }
  }

  setResultado(res);
};

const limpar = () => {
  setNum1('');
  setNum2('');
  setResultado(null);
};

return (
  <div >
    <h1>Calculadora Simples</h1>

    <div>
      <input
        type="number"
        value={num1}
        onChange={(e) => setNum1(e.target.value)}
        placeholder="Digite o Comprimento"
      />
      <input
        type="number"
        value={num2}
        onChange={(e) => setNum2(e.target.value)}
        placeholder="Digite sua Largura"
      />
    </div>

    <div>
      <button onClick={() => handleOperacao('Calcular a
area')}}>Calcular a area</button>

```

```

        <button onClick={limpar}
className="botao-limpar">Limpar</button>
      </div>

      {resultado !== null && (
        <h3 className="resultado">
          Resultado: {resultado}
        </h3>
      )}
    </div>
  );
}

export default Calculadora;

```

```

c)import { useState } from 'react';
import './App.css';

const taxasParaBRL = {
  BRL: 1.00,
  USD: 5.39,
  EUR: 6.25,
};

function ConversorDeMoedas() {
  const [valor, setValor] = useState('');
  const [moedaOrigem, setMoedaOrigem] = useState('BRL');
  const [moedaDestino, setMoedaDestino] = useState('USD');
  const [resultado, setResultado] = useState(null);

  const handleConversao = () => {
    const valorNum = parseFloat(valor);

    if (isNaN(valorNum) || valorNum <= 0) {
      setResultado('Erro: Insira um valor válido');
      return;
    }

    if (moedaOrigem === moedaDestino) {
      setResultado(`Resultado: ${valorNum.toFixed(2)}
${moedaDestino}`);
      return;
    }
  }
}

```

```
const valorEmBRL = valorNum * taxasParaBRL[moedaOrigem];
const valorFinal = valorEmBRL / taxasParaBRL[moedaDestino];

const resultadoFormatado = `${valorNum.toFixed(2)} ${moedaOrigem} =
${valorFinal.toFixed(2)} ${moedaDestino}`;
setResultado(resultadoFormatado);
};

const limpar = () => {
  setValor('');
  setResultado(null);
  setMoedaOrigem('BRL');
  setMoedaDestino('USD');
};

return (
  <div className="calculadora-container">
    <h1>Conversor de Moedas</h1>

    <div className="conversor-body">
      <div className="input-group">
        <label htmlFor="valor">Valor:</label>
        <input
          id="valor"
          type="number"
          value={valor}
          onChange={(e) => setValor(e.target.value)}
          placeholder="Digite o valor"
        />
      </div>

      <div className="input-group">
        <label htmlFor="origem">De:</label>
        <select
          id="origem"
          value={moedaOrigem}
          onChange={(e) => setMoedaOrigem(e.target.value)}
        >
          <option value="BRL">Real (BRL)</option>
          <option value="USD">Dólar (USD)</option>
          <option value="EUR">Euro (EUR)</option>
        </select>
      </div>
    </div>
  </div>
);
```

```

    </div>

    <div className="input-group">
      <label htmlFor="destino">Para:</label>
      <select
        id="destino"
        value={moedaDestino}
        onChange={ (e) => setMoedaDestino(e.target.value) }
      >
        <option value="BRL">Real (BRL)</option>
        <option value="USD">Dólar (USD)</option>
        <option value="EUR">Euro (EUR)</option>
      </select>
    </div>
  </div>

  <div className="botoes-container">
    <button onClick={handleConversao}
      className="botao-principal">Converter</button>
    <button onClick={limpar}
      className="botao-limpar">Limpar</button>
  </div>

  {resultado !== null && (
    <h3 className="resultado">
      {resultado}
    </h3>
  )}
</div>
);
}

export default ConversorDeMoedas;

```

```

11)a)import { useState } from 'react';
import './App.css';

```

```

function FormularioTempoReal() {
  const [dados, setDados] = useState({
    nome: '',
    email: '',
    idade: ''
  })
  const handleChange = (e) =>{

```

```

    const { name, value } = e.target;
    setDados({
      ...dados,
      [name]: value
    })
  }
}

return(
  <div>
    <h2>Formulário</h2>
    <form >
      <div>
        <label >Nome:</label>
        <input type="text"
          name='nome'
          value={dados.nome}
          onChange={handleChange} />
      </div>
      <div>
        <label >email:</label>
        <input type="email"
          name='email'
          value={dados.email}
          onChange={handleChange} />
      </div>
      <div>
        <label >idade:</label>
        <input type="number"
          name='idade'
          value={dados.idade}
          onChange={handleChange} />
      </div>
      <div>
        <h3>Dados preenchidos:</h3>
        <h5>seu nome é: {dados.nome}</h5>
        <h5>seu email é :{dados.email}</h5>
        <h5>sua idade é : {dados.idade}</h5>
      </div>
    </form>
  </div>
)
}

```

```
export default FormularioTempoReal;
```

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

```
import './App.css';
```

```
function FormularioTempoReal() {
```

```
  const [dados, setDados] = useState({
```

```
    nome: '',
```

```
    email: '',
```

```
    idade: ''
```

```
  });
```

```
  const [errors, setErrors] = useState({
```

```
    email: '',
```

```
    idade: ''
```

```
  });
```

```
  const handleChange = (e) => {
```

```
    const { name, value } = e.target;
```

```
    setDados({
```

```
      ...dados,
```

```
      [name]: value,
```

```
    });
```

```
  };
```

```
  const handleBlur = (e) => {
```

```
    const { name, value } = e.target;
```

```
    if (name === 'email') {
```

```
      if (!value.includes('@') && value !== '') {
```

```
        setErrors(prevErrors => ({
```

```
          ...prevErrors,
```

```
          email: 'O e-mail deve conter um "@".
```

```
        }));
```

```
      } else {
```

```
        setErrors(prevErrors => ({ ...prevErrors, email: '' }));
```

```
      }
```

```
    }
```

```
    if (name === 'idade') {
```

```
      const age = Number(value);
```

```
      if (age <= 18 && value !== '') {
```

```

        setErrors(prevErrors => ({
          ...prevErrors,
          idade: 'A idade deve ser maior que 18.'
        }));
      } else {
        setErrors(prevErrors => ({ ...prevErrors, idade: '' }));
      }
    }
  };

  return (
    <div>
      <h2>Formulário</h2>
      <form>
        <div>
          <label>Nome:</label>
          <input type="text"
            name='nome'
            value={dados.nome}
            onChange={handleChange} />
        </div>

        <div>
          <label>Email:</label>
          <input type="email"
            name='email'
            value={dados.email}
            onChange={handleChange}
            onBlur={handleBlur}
          />
          {errors.email && <span style={{ color: 'red', fontSize:
'12px' }}>{errors.email}</span>}
        </div>

        <div>
          <label>Idade:</label>
          <input type="number"
            name='idade'
            value={dados.idade}
            onChange={handleChange}
            onBlur={handleBlur}
          />

```

```

        {errors.idade && <span style={{ color: 'red', fontSize:
'12px' }}>{errors.idade}</span>}
      </div>

      <div>
        <h3>Dados preenchidos (em tempo real):</h3>
        <h5>Seu nome é: {dados.nome}</h5>
        <h5>Seu email é: {dados.email}</h5>
        <h5>Sua idade é: {dados.idade}</h5>
      </div>
    </form>
  </div>
);
}
export default FormularioTempoReal;

```

```

c)import { useState } from 'react';
import './App.css';

function FormularioTempoReal() {
  const [dados, setDados] = useState({
    nome: '',
    email: '',
    idade: '',
    cidade: '',
    estado: '',
    cep: ''
  });

  const [errors, setErrors] = useState({
    email: '',
    idade: ''
  });

  const handleChange = (e) => {
    const { name, value } = e.target;
    setDados({
      ...dados,
      [name]: value,
    });
  };
}

```



```

const handleBlur = (e) => {
  const { name, value } = e.target;

  if (name === 'email') {
    if (!value.includes('@') && value !== '') {
      setErrors(prevErrors => ({
        ...prevErrors,
        email: 'O e-mail deve conter um "@".'
      }));
    } else {
      setErrors(prevErrors => ({ ...prevErrors, email: '' }));
    }
  }

  if (name === 'idade') {
    const age = Number(value);
    if (age <= 18 && value !== '') {
      setErrors(prevErrors => ({
        ...prevErrors,
        idade: 'A idade deve ser maior que 18.'
      }));
    } else {
      setErrors(prevErrors => ({ ...prevErrors, idade: '' }));
    }
  }
};

return (
  <div>
    <h2>Formulário</h2>
    <form>
      <div>
        <label>Nome:</label>
        <input type="text"
          name='nome'
          value={dados.nome}
          onChange={handleChange} />
      </div>

      <div>
        <label>Email:</label>
        <input type="email"
          name='email'

```

```
        value={dados.email}
        onChange={handleChange}
        onBlur={handleBlur}
      />
      {errors.email && <span style={{ color: 'red', fontSize:
'12px' }}>{errors.email}</span>}
    </div>

    <div>
      <label>Idade:</label>
      <input type="number"
        name='idade'
        value={dados.idade}
        onChange={handleChange}
        onBlur={handleBlur}
      />
      {errors.idade && <span style={{ color: 'red', fontSize:
'12px' }}>{errors.idade}</span>}
    </div>
    <div>
      <label>estado:</label>
      <input type="text"
        name='estado'
        value={dados.estado}
        onChange={handleChange} />
    </div>
    <div>
      <label>Cidade:</label>
      <input type="text"
        name='cidade'
        value={dados.cidade}
        onChange={handleChange} />
    </div>
    <div>
      <label>CEP:</label>
      <input type="number"
        name='cep'
        value={dados.cep}
        onChange={handleChange} />
    </div>
    <div>
      <h3>Dados preenchidos (em tempo real):</h3>
      <h5>Seu nome é: {dados.nome}</h5>
    </div>
  </div>
</div>
```

```

        <h5>Seu email é: {dados.email}</h5>
        <h5>Sua idade é: {dados.idade}</h5>
        <h5>Seu estado é: {dados.estado}</h5>
        <h5>Sua cidade é: {dados.cidade}</h5>
        <h5>Seu CEP é: {dados.cep}</h5>
      </div>
    </form>
  </div>
);

```

```

}
export default FormularioTempoReal;

```

```

d)import { useState } from 'react';
import './App.css';

```

```

function AvatarCreator() {
  const [avatar, setAvatar] = useState({
    cabelo: 'Preto',
    olhos: 'Castanho',
    acessorios: []
  });

  const handleChange = (e) => {
    const { name, value } = e.target;
    setAvatar(prevAvatar => ({
      ...prevAvatar,
      [name]: value
    }));
  };

  const handleCheckboxChange = (e) => {
    const { value, checked } = e.target;

    setAvatar(prevAvatar => {
      const currentAcessorios = prevAvatar.acessorios;

      if (checked) {
        return {
          ...prevAvatar,
          acessorios: [...currentAcessorios, value]
        };
      } else {
        return {

```

```

        ...prevAvatar,
        acessorios: currentAcessorios.filter(item => item !== value)
    };
}
});
};

const acessoriosFormatados = avatar.acessorios.length > 0
    ? avatar.acessorios.join(', ')
    : 'Nenhum';

const isBia = avatar.cabelo === 'Castanho' &&
    avatar.olhos === 'Verde' &&
    avatar.acessorios.includes('Óculos');

const resultadoAvatar = isBia
    ? 'bia malheiros'
    : `👤 Cabelo: ${avatar.cabelo} | Olhos: ${avatar.olhos} |
Acessórios: ${acessoriosFormatados}`;

return (
    <div className="avatar-creator">
        <h2>Crie seu Avatar</h2>

        <div className="form-group">
            <label>Cor do Cabelo:</label>
            <select name="cabelo" value={avatar.cabelo}
onChange={handleChange}>
                <option value="Preto">Preto</option>
                <option value="Castanho">Castanho</option>
                <option value="Loiro">Loiro</option>
                <option value="Ruivo">Ruivo</option>
                <option value="Colorido">Colorido</option>
            </select>
        </div>

        <div className="form-group">
            <label>Cor dos Olhos (Radio):</label>
            <div className="radio-group">
                <label>
                    <input type="radio" name="olhos" value="Castanho"
checked={avatar.olhos === 'Castanho'}
onChange={handleChange} />

```

```

        Castanho
      </label>
      <label>
        <input type="radio" name="olhos" value="Azul"
          checked={avatar.olhos === 'Azul'}
onChange={handleChange} />
        Azul
      </label>
      <label>
        <input type="radio" name="olhos" value="Verde"
          checked={avatar.olhos === 'Verde'}
onChange={handleChange} />
        Verde
      </label>
    </div>
  </div>

  <div className="form-group">
    <label>Acessórios (Checkbox):</label>
    <div className="checkbox-group">
      <label>
        <input type="checkbox" value="Óculos"
          checked={avatar.acessorios.includes('Óculos')}
onChange={handleCheckboxChange} />
        Óculos
      </label>
      <label>
        <input type="checkbox" value="Chapéu"
          checked={avatar.acessorios.includes('Chapéu')}
onChange={handleCheckboxChange} />
        Chapéu
      </label>
      <label>
        <input type="checkbox" value="Colar"
          checked={avatar.acessorios.includes('Colar')}
onChange={handleCheckboxChange} />
        Colar
      </label>
    </div>
  </div>

  <div className="avatar-resultado">
    <h3>Seu Avatar :</h3>

```

```

        <p>
          {resultadoAvatar}
        </p>
      </div>
    </div>
  );
}

export default AvatarCreator;

```

```

12)a)import { useState } from 'react';
import './App.css';

function BotaoCurtir() {
  const [likes, setLikes] = useState(0);
  const [curtido, setCurtido] = useState(false);

  const toggleLike = () => {
    if (curtido) {
      setLikes(likes - 1);
      setCurtido(false);
    } else {
      setLikes(likes + 1);
      setCurtido(true);
    }
  };

  return (
    <button onClick={toggleLike}
      style={{
        backgroundColor: curtido ? '#1877f2' : '#e4e6ea',
        color: curtido ? 'white' : 'black',
        border: 'none',
        padding: '10px 20px',
        borderRadius: '5px',
        cursor: 'pointer'
      }}>

      {}
      👍 {curtido ? 'Curtido' : 'Curtir'} ({likes})

    </button>
  )
}

```

```
}  
export default BotaoCurtir;
```

```
b)import React, { useState } from 'react';  
import './App.css'  
function BotaoFavoritar() {  
  const [isFavoritado, setIsFavoritado] = useState(false);  
  const [contador, setContador] = useState(66);  
  
  const handleClick = () => {  
    if (isFavoritado) {  
      setContador(contador - 1);  
      setIsFavoritado(false);  
    } else {  
      setContador(contador + 1);  
      setIsFavoritado(true);  
    }  
  };  
  
  return (  
    <button  
      onClick={handleClick}  
      style={{  
        fontSize: '1.5rem',  
        padding: '10px 20px',  
        cursor: 'pointer',  
        border: '1px solid #ccc',  
        borderRadius: '8px'  
      }}  
    >  
      {isFavoritado ? '★' : '☆'}  
      {' '} {contador}  
    </button>  
  );  
}  
  
export default BotaoFavoritar;
```

```
c)import React, { useState } from 'react';  
import './App.css'  
function MedidorFelicidade() {  
  const [nivel, setNivel] = useState(0);
```

```

const expressoes = ['😐', '😊', '😄', '😁', '😆'];

const aumentarFelicidade = () => {
  if (nivel < expressoes.length - 1) {
    setNivel(nivel + 1);
  }
};

return (
  <div style={{ textAlign: 'center', padding: '20px' }}>

    <h1 style={{ fontSize: '6rem', margin: '10px' }}>
      {expressoes[nivel]}
    </h1>

    <p style={{ fontSize: '1.5rem', fontWeight: 'bold' }}>
      Nível de Felicidade: {nivel}
    </p>

    <button
      onClick={aumentarFelicidade}
      style={{
        fontSize: '1.2rem',
        padding: '12px 25px',
        cursor: 'pointer',
        borderRadius: '8px',
        border: 'none',
        backgroundColor: '#007bff',
        color: 'white'
      }}
      disabled={nivel === expressoes.length - 1}
    >
      Sorrir
    </button>

    {nivel === expressoes.length - 1 && (
      <p style={{ marginTop: '15px', color: '#007bff' }}>
        Felicidade máxima!
      </p>
    )}
  </div>
);
}

```



```
export default MedidorFelicidade;
```