

Atividade Final – React JS

Objetivo da Atividade

Avaliar sua compreensão dos principais conceitos de **React JS**, incluindo componentes, props, estado, ciclo de vida, eventos, formulários, modularização e estilização. A atividade contém parte teórica e parte prática.

Importante

- As respostas teóricas devem ser enviadas em um único arquivo `.pdf`, `.docx` ou `.txt`.
 - Os exercícios práticos devem ser enviados em arquivos separados, como `.jsx` ou `.tsx`, ou em pastas organizadas.
 - Você pode enviar tudo em `.zip` ou `.rar` ou publicar em um repositório no GitHub e enviar o link.
-

Instruções Gerais

- Utilize suas próprias palavras nas respostas teóricas.
 - Crie um arquivo por exercício prático (ou um projeto React organizado em pastas).
 - Inclua comentários explicando partes importantes do código.
 - Tudo pode ser entregue compactado ou como link do GitHub.
 - Não inclua explicação sobre critérios de nota.
-

PARTE 1 — Questões Teóricas

Responda todas as perguntas no arquivo teórico.

1. O que é **React** e por que ele é usado para criar interfaces de usuário?
 2. O que é uma **SPA (Single Page Application)**? Explique com suas palavras.
 3. O que é o **Virtual DOM** e por que ele melhora o desempenho das aplicações?
 4. O que são **componentes** em React e qual a vantagem de usá-los?
 5. O que são **props** e como elas permitem a comunicação entre componentes?
 6. Para que serve o hook `useState`? Explique seu funcionamento básico.
 7. Explique o que é o hook `useEffect` e dê um exemplo de uso comum.
 8. O que significa **lifting state up**? Dê um exemplo simples em palavras.
 9. O que caracteriza um **componente controlado** em React?
 10. Cite duas formas de **estilizar componentes React**.
 11. Por que é importante **modularizar** a estrutura de componentes em um projeto React?
-

PARTE 2 — Exercícios Práticos (React)

Os exercícios abaixo devem ser enviados cada um em arquivos separados, como `Exercicio1.jsx`, `Formulario.jsx`, etc. Você pode também criar um projeto em Vite ou Create React App e enviar a pasta completa compactada, ou publicar no GitHub e enviar o link.

Inclua comentários no código sempre que necessário.

♦ Exercício 1 — Componente Simples

Crie um arquivo `Exercicio1.jsx` contendo:

- Um componente funcional que exibe:
 - um `<h1>` com seu nome
 - um `<p>` falando brevemente sua experiência com React

♦ Exercício 2 — Uso de Props

Crie dois arquivos:

- `Mensagem.jsx` → recebe uma prop `texto` e exibe dentro de `<p>`
- `Exercicio2.jsx` → importa `Mensagem` e usa o componente duas vezes com textos diferentes

♦ Exercício 3 — Estado com `useState` (Contador)

Crie `Contador.jsx` com:

- estado `numero`, começando em 0
- botão `+1` (incrementa)
- botão `-1` (decrementa)
- exibição do valor no JSX

♦ Exercício 4 — Formulário Controlado

Crie `Formulario.jsx` com:

- um `input` controlado usando `useState`
- um botão que, ao clicar, mostra o valor digitado no console

♦ Exercício 5 — `useEffect` com Intervalo

Crie `Timer.jsx` contendo:

- estado `tempo` iniciando em 0
- `useEffect` para incrementar o tempo a cada 1 segundo
- `return` do `useEffect` deve limpar o intervalo
- exibir no JSX: “Tempo: X segundos”

♦ Exercício 6 — Lifting State Up

Crie:

- `InputNome.jsx`: `input` controlado, chama função enviada por `props` ao mudar texto
- `ExibirNome.jsx`: recebe prop `nome`, exibe: “Olá, nome!”
- `Exercicio6.jsx`: mantém estado `nome`, envia o valor e o `setter` para os componentes filhos

♦ Exercício 7 — Lista de Tarefas

Crie `ListaTarefas.jsx` com:

- `input` controlado
- botão “Adicionar”
- array de tarefas no estado
- renderização dinâmica usando `.map()`

♦ Exercício 8 — Estilização em React

Crie dois arquivos:

- `Exercicio8.jsx`
- `Exercicio8.module.css`
- Demonstre:
 - uso de **CSS Module**
 - uso de `className`
 - estilização básica (cor, borda, padding, etc.)

♦ Exercício 9 — Projeto Final (Mini Aplicação React)

Crie uma pasta chamada `projeto-final-react` contendo uma pequena aplicação com:

Requisitos mínimos:

- pelo menos 3 componentes funcionais reutilizáveis
- uso de `useState`

- uso de `useEffect`
- um formulário controlado
- interação entre componentes (`props` ou `lifting state up`)
- estilização organizada (CSS, module CSS ou inline)

Exemplos de temas possíveis:

- lista de produtos
- mini agenda de contatos
- app de tarefas
- lista de notas
- painel de contador avançado
- mini dashboard simples

A aplicação pode ser simples, mas deve demonstrar domínio dos conceitos estudados.

✓ ENTREGA

Você pode enviar a atividade de uma das formas abaixo:

✓ 1. Enviar arquivo compactado

📦 Envie um `.zip` ou `.rar` contendo:

- arquivo teórico
- todos os arquivos `.jsx` e `.css` dos exercícios
- pasta do projeto final

✓ 2. Repositório no GitHub

📌 Publique a atividade em um repositório e envie apenas o link.