**Atividade React**

**Aluno:** Geovane de Lucas Quinhone Ker **RA:** 14511 **Turma:** 4ºA

**1. O que são hooks no React e por que eles foram criados?**

R: Os hooks permitem usar vários recursos de forma mais simples através de funções, também ajuda a organizar a lógica usada dentro dos componentes. Foram criados para permitir o uso de estado e recursos do React em componentes funcionais, evitando precisar de classes, fazendo com que o código seja mais simples, legível e fácil manuseio.

**2. Explique o funcionamento dos hooks:**

R:

* useState: permite a criação de estado no componente através de função mantém o gerenciamento do estado local do componente, retornando um array como resultado.
* useEffect: auxilia a lidar com os side-effects (efeitos colaterais) e podemos usar também como ciclo de vida do componente.
* useCallback: serve para memorizar funções, ele garante que a mesma função seja reutilizada entre renders(renderização), assim evita que ela seja recriada toda vez que o componente atualizar.
* useRef: cria uma referência mutável que persiste entre os renders do componente, assim não causa uma nova renderização quando o seu valor muda.

**3. Cite pelo menos um exemplo prático de situação em que cada hook pode ser usado.**

* useState: Controle de um contador em um botão:

import { useState } from 'react';

function Contador() {

const [count, setCount] = useState(0);

return (

<div>

<p>Contador: {count}</p>

<button onClick={() => setCount(count + 1)}>Incrementar</button>

</div>

);

}

* useEffect: Buscar dados de um servidor quando o componente é montado.

import { useState, useEffect } from 'react';

function Usuarios() {

const [users, setUsers] = useState([]);

useEffect(() => {

fetch('https://jsonplaceholder.typicode.com/users')

.then(res => res.json())

.then(data => setUsers(data));

}, []);

return (

<ul>

{users.map(user => (

<li key={user.id}>{user.name}</li>

))}

</ul>

);

}

* useCallback: Evitar que um componente filho renderize desnecessariamente.

import { useState, useCallback } from 'react';

function ContadorFilho({ increment }) {

console.log('Filho renderizou');

return <button onClick={increment}>Incrementar</button>;

}

function Pai() {

const [count, setCount] = useState(0);

const increment = useCallback(() => {

setCount(c => c + 1);

}, []);

return (

<div>

<p>Contador: {count}</p>

<ContadorFilho increment={increment} />

</div>

);

}

* useRef: Foca automaticamente em um input.

import { useRef, useEffect } from 'react';

function Formulario() {

const inputRef = useRef(null)

useEffect(() => {

inputRef.current.focus();

}, []);

return <input ref={inputRef} placeholder="Digite aqui chinês..." />;

}