**Ejercicios sesiones 4, 5 y 6**

Partiendo del siguiente componente de clase que contempla varios métodos de ciclo de vida, convertidlo en un componente funcional que realice exactamente lo mismo:

import React, { Component } from 'react'

import ‘../../styles/clock.scss’;

class Clock extends Component {

constructor(props) {

super(props);

// Estado privado del component

this.state = {

// Se genera una fecha como estado inicial del componente

fecha: new Date(),

edad: 0,

nombre: ‘Martín’,

apellidos: ‘San José’

};

}

componentDidMount(){

this.timerID = setInterval (

() => this.tick(), 1000

);

}

componentWillUnmount() {

clearInterval (this.timerID);

}

render() {

return (

<div>

<h2>

Hora Actual:

{this.state.fecha.toLocaleTimeString()}

</h2>

<h3>{this.state.nombre} {this.state.apellidos}</h3>

<h1>Edad: {this.state.edad}</h1>

</div>

)

}

tick(){

this.setState((prevState) => {

let edad = prevState.edad +1;

return {

...prevState,

fecha: new Date(),

edad

}

});

}

}

export default Clock;

**En un componente funcional (NO puedo verificarlo si esta bien. Muchas dudas)**

import React, {useState, useEffect} from 'react';

const Clock = () => {

const [fecha, setFecha] = useState(fecha);

const [edad, setEdad] = useState(0);

const [nombre, setNombre = useState(‘Martín’);

const [apellidos, setApellidos] = useState(‘San José’);

useEffect(() => {

    console.log(persona);

    const timerID = setInterval(() =>{

        tick(), 1000);

    return () => {

        console.log('Componente va a desaparecer');

        clearInterval(timerID));

    };

   }, []);

    return (

        <div>

<h2>{fecha}</h2>

<h3>{nombre}</h3>

<h3>{apellidos}</h3>

<h1>Edad: {edad}</h1>

</div>

)

}

tick(){

this.setState((prevState) => {

let edad = prevState.edad +1;

return {

...prevState,

fecha: new Date(),

edad

}

});

        </div>

    );

}

export default Clock;