**REACT HOOKS**

**PARA CLASES Y FUNCIONES**

(para funciones todos se hacen con el hook useEffect)

**componentDidMount()**

/\*\*

\* Ejemplo de uso del método

\* de ciclo de vida en componete clase y el hook de ciclo de vida

\* en componente funcional

\*/

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

export class Didmount extends Component {

componentDidMount(){

console.log('Comportamiento antes de que el componente sea añadido al DOM (rederice)');

}

render() {

return (

<div>

<h1>DidMount</h1>

</div>

);

}

}

export const DidmountHook = () => {

useEffect(() => {

console.log('Comportamiento antes de que el componente sea añadido al DOM (rederice)');

}, [])

**//** **AL USAR EL CORCHETE VACIO SE EJECUTA UNA SOLA VEZ Y NO USAMOS EL**

**// RETURN DE USEEFECT()**

return (

<div>

<h1>DidMount</h1>

</div>

);

}

**componentDidUpdate()**

/\*\*

\* Ejemplo de uso de método componentDidUpdate en componente de clase

\* y uso de hook en componete funcional

\*/

import React, { Component, useEffect } from 'react'

export class DidUpdate extends Component {

componentDidUpdate(){

console.log('Comportamiento cuando el componente recibe nuevos props o cambio en su estado privado')

}

render() {

return (

<div>

<h1>DidUpdate</h1>

</div>

)

}

}

export const DidupdateHook = () => {

useEffect(() => {

console.log('Comportamiento cuando el componente recibe nuevos props o cambio en su estado privado')

});

**// AL NO TENER CORCHETES FINALES, SE EJECUTA CADA VEZ QUE CAMBIA**

**//** **EL ESTADO DEL COMPONENTE Y NO USAMOS EL RETURN DE USEEFFECT()**

return (

<div>

<h1>DidUpdate</h1>

</div>

);

}

**componentWillUnmount()**

/\*\*

\* Ejemplo de uso del método componentWillUnmount para componente clase

\* Ejemplo de uso del hooks para componente funcional

\* (Cuando el componente va a desaparecer)

\*/

import React, { Component, useEffect } from 'react'

export class WillUnMount extends Component {

componentWillUnmount(){

console.log('Comportamiento antes de que el componente desaparezca');

}

render() {

return (

<div>

<h1>

WillUnmount

</h1>

</div>

)

}

}

export const WillunmountHook = () => {

useEffect(() => {

// aqui no ponemos nada

return () => {

console.log('Comportamiento antes de que el componente desaparezca');

};

}, []);

**// SE EJECUTARA UNA SOLA VEZ, POR TENER LOS CORCHETES… Y UTILIZARA EL**

**// RETURN DEL USEEFFECT()**

return (

<div>

<h1>

WillUnmount

</h1>

</div>

);

}