

## SEMANA 10

Crearemos un proyecto en VITE

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.2965]
(c) Microsoft Corporation. Todos los derechos reservados.

E:\DATA C\Downloads\IX Desarrollo web\Laboatorios\Semana9 Y 10>npm create vite proydemo1
Need to install the following packages:
create-vite@5.2.3
Ok to proceed? (y) y
? Select a framework: » - Use arrow-keys. Return to submit.
  Vanilla
  Vue
>  React
  Preact
  Lit
  Svelte
  Solid
  Qwik
  Others
```

Seleccionamos las siguientes opciones

```
E:\DATA C\Downloads\IX Desarrollo web\Laboatorios\Semana9 Y 10>npm create vite proydemo1
Need to install the following packages:
create-vite@5.2.3
Ok to proceed? (y) y
✓ Select a framework: » React
✓ Select a variant: » JavaScript

Scaffolding project in E:\DATA C\Downloads\IX Desarrollo web\Laboatorios\Semana9 Y 10\proydemo1...

Done. Now run:

  cd proydemo1
  npm install
  npm run dev

npm notice
npm notice New minor version of npm available! 10.5.2 -> 10.8.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.1
npm notice Run npm install -g npm@10.8.1 to update!
npm notice

E:\DATA C\Downloads\IX Desarrollo web\Laboatorios\Semana9 Y 10>cd proydemo1

E:\DATA C\Downloads\IX Desarrollo web\Laboatorios\Semana9 Y 10\proydemo1>npm install
[ ] | idealTree:proydemo1: sill idealTree buildDeps
```

Dentro de la terminal para ejecutar el proyecto "npm run dev"

```
run `npm fund` for details

found 0 vulnerabilities

E:\DATA C\Downloads\IX Desarrollo web\Laboatorios\Semana9 Y 10\proydemo1>npm i react-router-dom
[ ] | idealTree:proydemo1: sill idealTree buildDeps
```

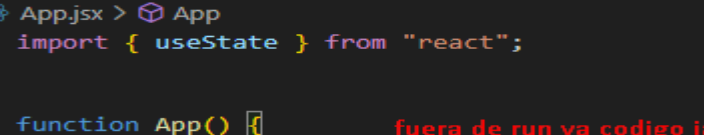
### CRACION DE USSTATE

Cada vez que quiero usar el código de useState usamos el disparador use (necesitas instalar antes).

Nombre del useState "name" ello es como una variable se puede renderizar

```
const [name, setName] = useState();
```

Si dentro de tu proyecto existe códigos de color rojo es porque no está importado.



```
index.html App.jsx App.css index.css
src > App.jsx > App
1 import { useState } from "react";
2
3
4 function App() {
5     // fuera de run va codigo javascript
6     const [name, setName] =useState();
7     return (
8         // aquí html y css
9         <>
10         |   <h2>Usando uState</h2>
11         |   </>
12         )
13     }
14
15 export default App
16
```

De esta manera mostramos el UseState

```


src > App.jsx > App
1  import { useState } from "react";
2
3
4  function App() {
5
6      const [name, setName] =useState("");
7      const [address, setAddress]=useState("");
8      return (
9
10         <>
11             <h2>Usando uState</h2>
12             <p>nmbre: {name}</p>
13             <p>direccion: {address}</p>
14         </>
15     )
16 }
17
18 export default App
19

```

También utilizamos eventos de un botón a partir de `OnClick`.

El Sniped MET crea otra estructura rápida.

```

const [name, setName] =useState("DELMIZA");
const [address,setAddress]=useState("ESQUIVEL");
//creando funcion para boton
const met
return (  meta Emmet

```

Aquí podemos visualizar la función que se le dará al botón.

```

index.html  App.jsx 1  App.css  index.css
src > App.jsx > App > handleClick
1  import { useState } from "react";
2
3
4  function App() {
5
6      const [name, setName] =useState("DELMIZA");
7      const [address,setAddress]=useState("ESQUIVEL");
8      //creando funcion para boton
9      const handleClick =() => {third}
10     return (
11
12         <>
13             <h2>Usando uState</h2>
14             <p>nombre: {name}</p>
15             <p>direccion: {address}</p>
16             <button onClick={}>Cambiar</button>
17         </>
18     )
19 }
20
21 export default App
22

```

De esta forma no puedes manipular al Usestate tienes que utilizar su set

```

index.html  App.jsx  App.css  index.css
src > App.jsx > App > handleClick
1  import { useState } from "react";
2
3
4  function App() {
5
6      const [name, setName] =useState("DELMIZA");
7      const [address,setAddress]=useState("ESQUIVEL");
8      //creando funcion para boton
9      const handleClick =() => {
10         name = "Jamime profe"
11     }
12     return (
13
14         <>
15             <h2>Usando uState</h2>
16             <p>nombre: {name}</p>
17             <p>direccion: {address}</p>
18             <button onClick={handleClick}>Cambiar</button>
19         </>
20     )
21 }
22
23 export default App
24

```

De esta forma es correcta

```
index.html App.jsx App.css index.css
src > App.jsx > App
1 import { useState } from "react";
2
3
4 function App() {
5
6   const [name, setName] =useState("DELMIZA");
7   const [address, setAddress]=useState("ESQUIVEL");
8   //creando funcion para botton
9   const handleClick =() => {
10     setName("Jaime")
11     setAddress("no vive")
12   }
13   return (
14
15     <>
16     <h2>Usando uState</h2>
17     <p>nombre: {name}</p>
18     <p>direccion: {address}</p>
19     <button onClick={handleClick}>Cambiar</button>
20   )
21 )
22 }
23
24 export default App
25
```

## USANDO USEEFFECT

Sniped UseEffect y crea estructura

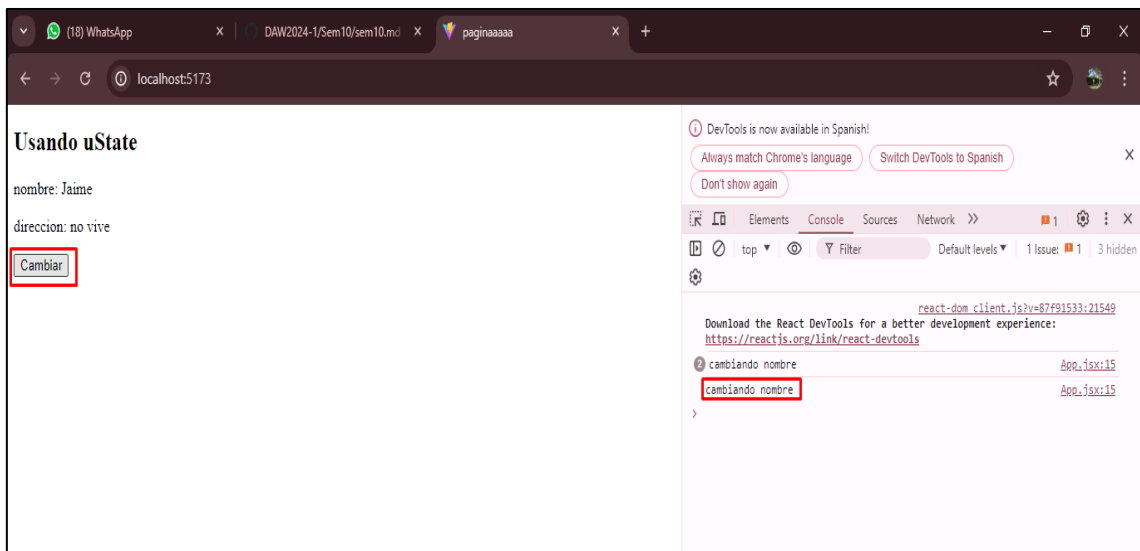
```
//Pertenece Useeffect
useEffect(() => {
  first
  return() => {
    second
  }
},[third])
return (
```

En esta parte se pone los UseState tu puedes poner cualquier UseState.

```
//Pertenece Useeffect
useEffect(() => {
  console.log("cambiando nombre")
},[name])
return (
```

**cambia en esta parte el name**

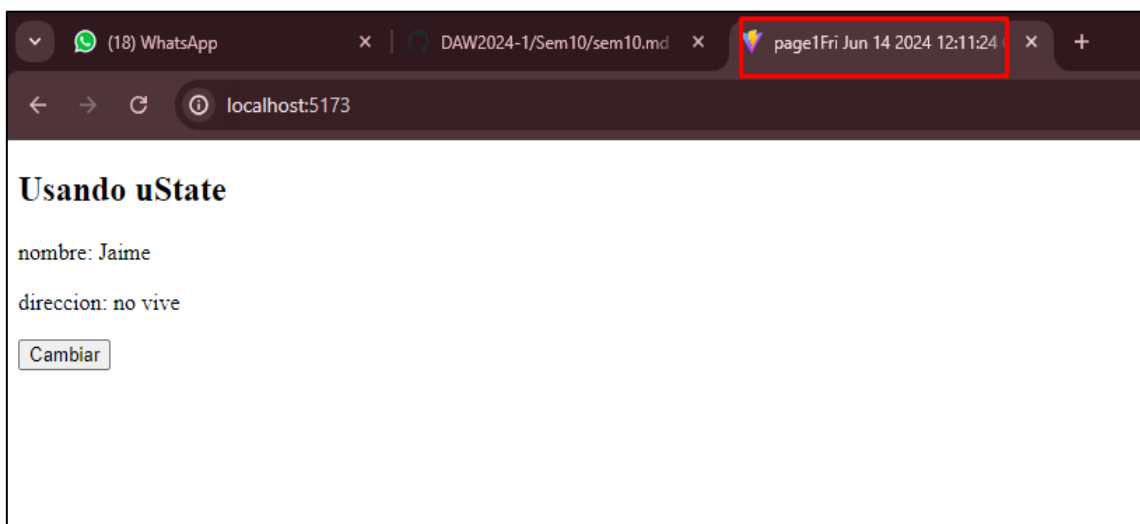
En esta imagen se puede observar la ejecución de los UseEffect



Este es otro ejemplo del uso de UseEffect

```
//Pertenece Useeffect
useEffect(() => {
  let fecha = new Date();
  console.log("cambiando nombre")
  document.title="page1" + fecha
}, [name])
```

Ejecución del uso de hook



Puedo eliminar lo que está dentro de los corchetes, pero en la consola ya no se mostrará cuando hago ese cambio haciendo el click.

```
index.html App.jsx App.css index.css
src > App.jsx > App
1 import { useEffect, useState } from "react";
2
3
4 function App() {
5   //PERTENECE A USE STATE
6   const [name, setName] =useState("DELMIZA");
7   const [address, setAddress]=useState("ESQUIVEL");
8   //creando funcion para boton
9   const handleClick =() => {
10     setName("Jaime")
11     setAddress("no vive")
12   }
13   //Pertenece Useeffect
14   useEffect(() => {
15     let fecha = new Date();
16     console.log("cambiando nombre")
17     document.title="page1" + fecha
18   },[])
```

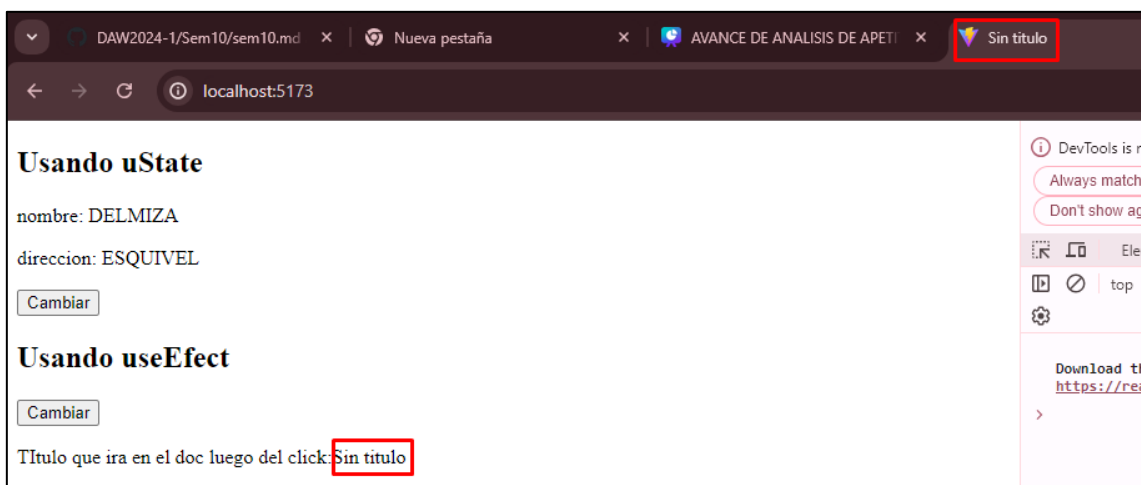
UseEffect ejemplo de git

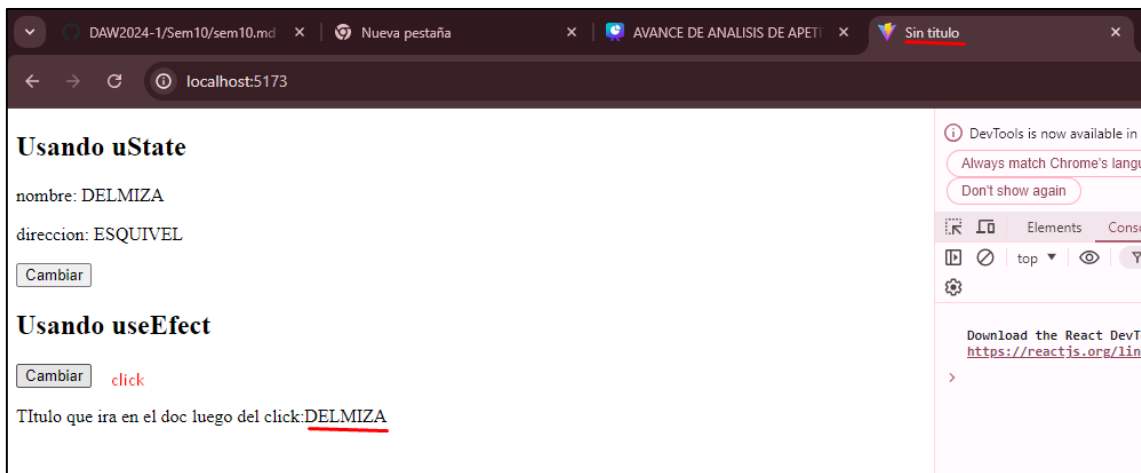
En este caso si sacamos de los corchetes título no se cambiará al hacer click

```
//Pertenece a useeffect
const [titulo, setTitulo] =useState ("Sin titulo");
useEffect (() => {
  document.title=titulo
},[])

const handleClick2 =() => {
  setTitulo(name)
}
```

Ejemplo de ejecución del UseState y UseEffect

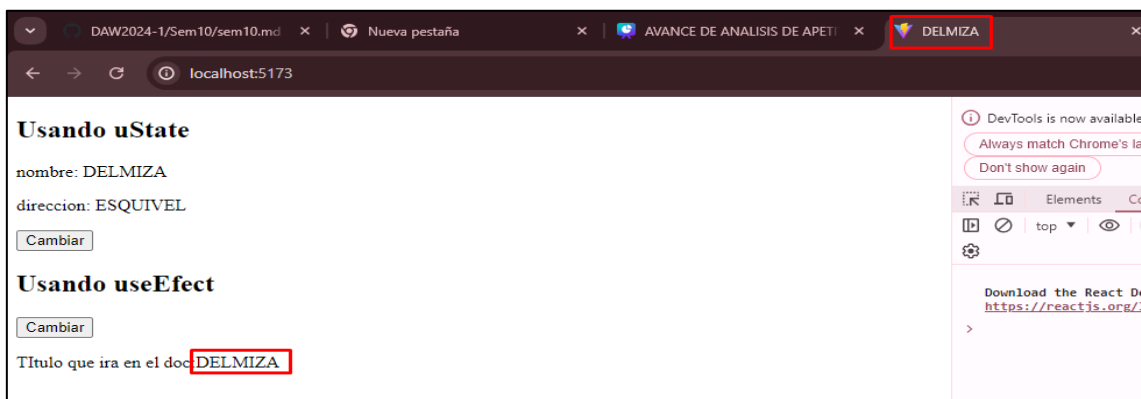




Si ingresamos el título dentro de los corchetes si hace cambio

```
//Pertenece a Useeffecto
const [titulo, setTitulo] =useState ("Sin titulo");
useEffect (() => {
  | document.title=titulo
},[titulo])

const handleClick2 =() => {
  | setTitulo(name)
}
```



## USECONTEXT

Creo dos componentes en le padre están los datos y en el hijo mostramos

```
src > components > Padre.jsx > ...
1  import React from 'react'
2  import Hijo from './Hijo'
3
4  // export default function Padre() {
5  //   return (
6  //     <div>Padre holi</div>
7  //   )
8  // }
9  export const MyContext=React.createContext()
10 const Padre = () =>{
11   let docente = {
12     nombre: "delmi",
13     apellido: "esquivel"
14   }
15   return (
16     <MyContext.Provider value={docente}>
17       <Hijo/>
18     </MyContext.Provider>
19   )
20 }
21 export default Padre;
22
```

```
src > components > Hijo.jsx > [default]
1  import React, { useContext } from 'react'
2  import { MyContext } from './Padre'
3
4  // export default function Hijo() {
5  //   return (
6  //     <div>Hijo</div>
7  //   )
8  // }
9
10 const Hijo = () => {
11   const dato =useContext(MyContext)
12   return(
13     <div>
14       <p>{dato.nombre}</p>
15       <p>{dato.apellido}</p>
16     </div>
17   )
18 }
19 export default Hijo;
```

No olvides los subrayados de rojo export default

Y dentro de App.jsx agregar los componentes.

```
src > App.jsx > ...
1  import { Component, useEffect, useState } from "react";
2  import Padre from "../components/Padre";
3
4  function App() {
5    //Pertenece al UseSTATE
```



```

return [
  <>
    <h2>Usando useState</h2>
    <p>nombre: {name}</p>
    <p>direccion: {address}</p>
    <button onClick={handleClick}>Cambiar nombres</button>
    <h2>Usando useEfect</h2>
    <button onClick={handleClick2}>Cambiar titulo</button>
    <p>Titulo que ira en el doc luego del click:{titulo}</p>
    <h2>USANDO useContext</h2>
    <div>
      <Padre/>
    </div>
  </>
]

```

## CUSTOM HOOK

UseFetch consumir sirve para consumir API (jsonplaceholder nos proporciona APIs para poder trabajar.

Leer a travez de fetch y hook personalizado.

Crear dos componentes no olvides importar si usas UseEffect y useState además de export default en caso arriba no se haga.

```

index.html App.jsx Padre3.jsx EfectoFetch.jsx Hijo.jsx Padre.jsx
src > components > EfectoFetch.jsx > ...
1 import React, { useEffect, useState } from 'react'
2
3 //Pertenece a use Custom Hook
4 const EfectoFetch = (url) =>{
5   const [data, setData] = useState(null);
6   useEffect(() =>{
7     fetch(url)
8       .then((res)=>res.json())
9       .then((data) =>setData(data));
10   },[url]);
11   return[data];
12 };
13
14 export default EfectoFetch;
15

```

```

index.html App.jsx Padre3.jsx EfectoFetch.jsx Hijo.jsx Padre.jsx
src > components > Padre3.jsx > default
1 import React from 'react'
2 import EfectoFetch from './EfectoFetch';
3
4
5 // export default function Padre3() { cuando no esta este expor se pone el rojo de
6 //   return ( abajo
7 //     <div>Padre3</div>
8 //   )
9 // }
10
11 const Padre3 =()=>{
12   const [data] = EfectoFetch("https://jsonplaceholder.typicode.com/todos");
13   return (
14     <>
15       {data &&
16         data.map((item) => {
17           return <p key={item.id}>Title: {item.title}</p>;
18         })}
19     </>
20   );
21 }
22
23 export default Padre3;

```

Y en el App importamos

```
index.html App.jsx Padre3.jsx EfectoFetch.jsx
src > App.jsx > ...
1 import { Component, useEffect, useState } from "react";
2 import Padre from "../components/Padre";
3 import Padre3 from "../components/Padre3";
```

```
return (
  <>
    <h2>1 Usando useState</h2>
    <p>nombre: {name}</p>
    <p>direccion: {address}</p>
    <button onClick={handleClick}>Cambiar nombres</button>
    <h2>2 Usando useEffect</h2>
    <button onClick={handleClick2}>Cambiar titulo</button>
    <p>Titulo que ira en el doc luego del click:{titulo} </p>
    <h2>3 Usando useContext</h2>
    <div>
      <Padre/>
    </div>
    <h2>4 Usando Custom Hook</h2>
    <div>
      <Padre3/>
    </div>
  </>
)
```

## Hook USEREDUCE

### EXPLICACION EJERCICOS

Ojo: Si estás exportando un objeto literal o una función, asegúrate de no usar export default si no estás exportando un solo objeto o función. En su lugar, usa la palabra clave export seguida del nombre de cada const