

REACT - Hook

En React encontramos diferentes componentes funcionales para el manejo de datos y el estado de ellos. Entre estas funciones especiales (conocidas como hook) implementaremos UseEffect y UseState.

UseState

Se utiliza para guardar y actualizar valores que pueden cambiar en el tiempo dentro del componente. Estos valores pueden ser: datos de formularios, lista de usuarios, contadores, estado de botones entre otros.

Ej: `const [contador, setContador] = useState(0);`

Este hook crea una variable llamada contador y la función setContador se utiliza para actualizarlo. Cada vez que se actualiza el componente se vuelve a renderizar.

UseEffect

Se utiliza para ejecutar efectos secundarios, es decir, codificación que no está relacionado con el renderizado, por ej:

- Llamadas a APIs
- Manipular el DOM
- Temporizadores
- Almacenamiento en LocalStorage

Ej:

```
useEffect(() => {
    console.log("El componente se montó");
}, []);
```

Esto se ejecuta una vez, al haber iniciado el ciclo del componente.

Ej:

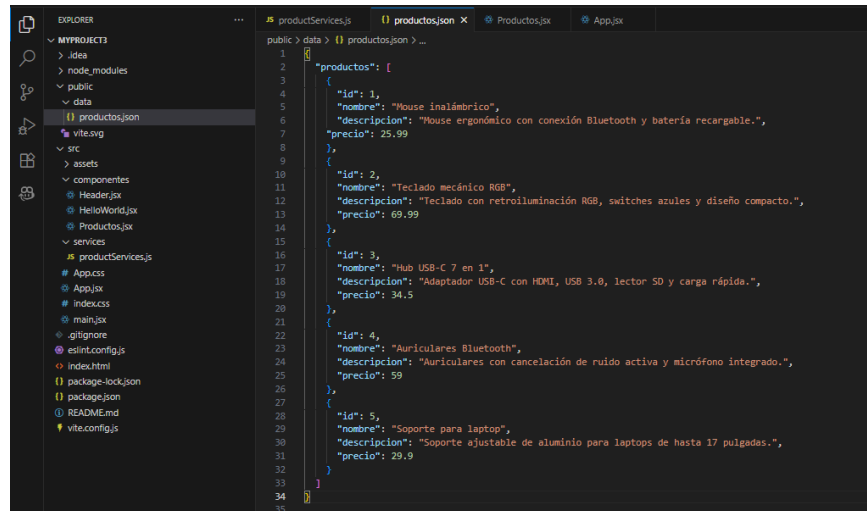
```
const [usuarios, setUsuarios] = useState([]);

useEffect(() => {
    fetch("https://api.example.com/usuarios")
        .then(res => res.json())
        .then(data => setUsuarios(data));
}, []);
```

useState: almacena los usuarios en el estado.

useEffect: carga la información desde una API al ejecutar un componente.

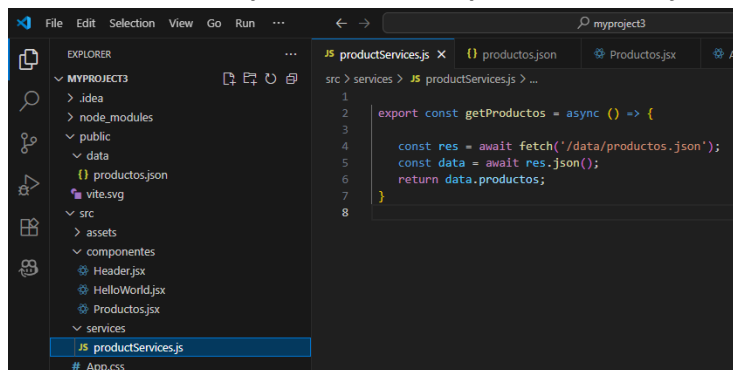
Implementaremos un archivo productos.json almacenado en la carpeta /public/data:



```
1  {} productos.json > ...
2
3  "productos": [
4    {
5      "id": 1,
6      "nombre": "Mouse inalámbrico",
7      "description": "Mouse ergonómico con conexión Bluetooth y batería recargable.",
8      "precio": 25.99
9    },
10   {
11     "id": 2,
12     "nombre": "Teclado mecánico RGB",
13     "description": "Teclado con retroiluminación RGB, switches azules y diseño compacto.",
14     "precio": 69.99
15   },
16   {
17     "id": 3,
18     "nombre": "Hub USB-C 7 en 1",
19     "description": "Adaptador USB-C con HDMI, USB 3.0, lector SD y carga rápida.",
20     "precio": 34.5
21   },
22   {
23     "id": 4,
24     "nombre": "Auriculares Bluetooth",
25     "description": "Auriculares con cancelación de ruido activa y micrófono integrado.",
26     "precio": 59
27   },
28   {
29     "id": 5,
30     "nombre": "Soporte para laptop",
31     "description": "Soporte ajustable de aluminio para laptops de hasta 17 pulgadas.",
32     "precio": 29.9
33   }
34 ]
35
```

Ejecutamos el servidor de json: /public/data/ json-server productos.json

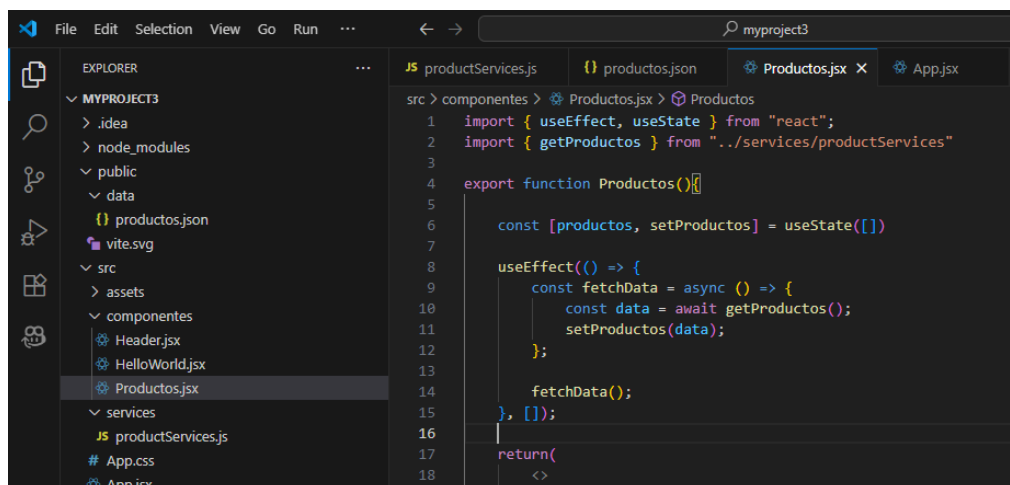
Crearemos en la carpeta src/services: productServices.js



```
1
2
3  export const getProductos = async () => {
4
5    const res = await fetch('/data/productos.json');
6    const data = await res.json();
7    return data.productos;
8  }
```

Implementamos método async()
getProductos a través del método
fetch()

Ahora creamos el componente: src/componentes/Productos.jsx



```
1  import { useEffect, useState } from "react";
2  import { getProductos } from "../services/productServices"
3
4  export function Productos()
5
6    const [productos, setProductos] = useState([])
7
8    useEffect(() => {
9      const fetchData = async () => {
10        const data = await getProductos();
11        setProductos(data);
12      };
13
14      fetchData();
15    }, []);
16
17    return(
18      <>
```

```
const [productos, setProductos] = useState([])
```

- Declaramos un estado local llamado productos.
- useState([]) inicia un array vacío
- productos es el parámetro que contiene el estado actual
- setProductos es la función que permite actualizar el estado del producto.

```
useEffect(() => {
```

```
  const fetchData = async () => {  
    const data = await getProductos();  
    setProductos(data);  
  };  
  
  fetchData();  
}, []);
```

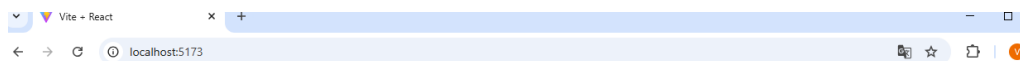
- useEffect() Es un hook que permite renderizar getProductos() del service.
- Se define una función fetchData la que espera el retorno de la función getProductos()
- Actualiza el estado de productos a través de setProductos(data)
- Posterior a eso, se llama o invoca a la función.

Posterior a esto, retornamos el mapeo de productos en código web:

```
return(  
  <>  
    <div className="container">  
      <h3>Lista de Productos</h3>  
      <div className="row">  
        {productos.map(prod =>(  
          <div className="col-md-4 my-2" key={prod.id}>  
            <div className="card">  
              <div className="card-body">  
                <h4 className="card-title">{prod.nombre}</h4>  
                <p className="card-text">{prod.descripcion}</p>  
                <p className="card-text">{prod.precio}</p>  
              </div>  
            </div>  
          </div>  
        )) }  
      </div>  
    </div>  
  </>  
)
```

Finalmente, el componente debe visualizarse de la siguiente forma:

```
src > componentes > Productos.jsx > useEffect() callback > fetchData
1 import { useEffect, useState } from "react";
2 import { getProductos } from "../services/productServices"
3
4 export function Productos(){
5
6     const [productos, setProductos] = useState([])
7
8     useEffect(() => {
9         const fetchData = async () => {
10             const data = await getProductos();
11             setProductos(data);
12         };
13         fetchData();
14     }, []);
15
16     return(
17         <>
18         <div className="container">
19             <h3>Lista de Productos</h3>
20             <div className="row">
21                 <div className="col-md-4 my-2" key={prod.id}>
22                     <div className="card">
23                         <div className="card-body">
24                             <h4 className="card-title">{prod.nombre}</h4>
25                             <p className="card-text">{prod.descripcion}</p>
26                             <p className="card-text">{prod.precio}</p>
27                         </div>
28                     </div>
29                 </div>
30             </div>
31         </div>
32     </>
33 )
34 }
35
36
37
38 }
```



Lista de Productos

Mouse inalámbrico

Mouse ergonómico con conexión Bluetooth y batería recargable.

25,99

Teclado mecánico RGB

Teclado con retroiluminación RGB, switches azules y diseño compacto.

69,99

Hub USB-C 7 en 1

Adaptador USB-C con HDMI, USB 3.0, lector SD y carga rápida.

34,5

Auriculares Bluetooth

Auriculares con cancelación de ruido activa y micrófono integrado.

59

Soporte para laptop

Soporte ajustable de aluminio para laptops de hasta 17 pulgadas.

29,9