

## APUNTES REACT

Abel Rios - 17/05/2022

### \*\*\*EJERCICIO 2\*\*\*

#### Ejercicio 2

1. Creamos un nuevo proyecto React.
2. Creamos un componente llamado Tienda.js que tendrá un objeto como el que se muestra en la imagen:

```
const tienda = {  
  electronica: [  
    {id: 27, producto: "Televisor", marca: "LG", modelo: "XP7302", precio: 399},  
    {id: 28, producto: "Equipo Hi-FI", marca: "Samsung", modelo: "VF235", precio: 179},  
    {id: 29, producto: "Televisor", marca: "Sony", modelo: "Bravia-173", precio: 498},  
  ],  
  alimentacion: [  
    {id: 30, producto: "Galletas", marca: "María", precio: 0.90},  
    {id: 31, producto: "Ensalada", marca: "Imizurra", precio: 1.30},  
    {id: 32, producto: "Patatas", marca: "McKain", precio: 0.90},  
    {id: 33, producto: "Pasta", marca: "Gallo", precio: 0.90},  
  ],  
  mascotas: [  
    {id: 34, producto: "Croquetas para gato", marca: "Purina", precio: 4.90},  
    {id: 35, producto: "Arena de gato", marca: "Limpior", precio: 1.10},  
  ]  
};
```


#### Ejercicio 2 (cont.)

3. Creamos un nuevo componente llamado Lista.js y lo renderizamos tres veces en Tienda.js.
4. A las listas les pasamos un atributo con el nombre de cada categoría y otro con los productos correspondientes.
5. En la extensión del navegador, comprobamos que los componentes reciben los props.
6. Lista tendrá un título con el nombre de la categoría y tantos list items como productos reciba.

Creamos nuestro componente Tienda y guardamos el objeto tienda tocho en él (aunque debería venir de la base de datos). Para empezar vamos a desestructurar el objeto tienda en tres objetos que sean las distintas categorías:

```
20 const {electronica,mascotas,alimentacion} = tienda  
21 // Desestructuramos el objeto tienda en tres objetos: electronica, mascotas y alimentación  
22 // estos objetos los podemos usar ahora sin necesidad de decir tienda.electronica (por ejemplo)  
23
```

Ahora creamos el componente Lista, que será un map de los arrays (categorías) pintados como ul y li.

```
ejercicio2 > src > components > Lista > JS Lista.js >  Lista  
1 export default function Lista({ productos , categoria }) {  
2 // Recibe dos objetos distintos, pero van dentro de la misma llave  
3
```

Lista.js (le añadimos la key para que no nos de el warning en la consola)

```
return (  
  <>  
    <h1>{categoria}</h1>  
    {  
      productos.map((item) => {  
        return (  
          <ul key={item.id}>  
            <li>{item.id}</li>  
            <li>{item.producto}</li>  
            <li>{item.marca}</li>  
            <li>{item.modelo}</li>  
            <li>{item.precio}</li>  
          </ul>  
        )  
      })  
    }  
  </>  
)
```

Ahora en nuestro componente Tienda.js importamos Lista.js

```
ejercicio2 > src > components > Tienda > JS Tienda.js > ...  
1   import Lista from "../Lista/Lista"
```

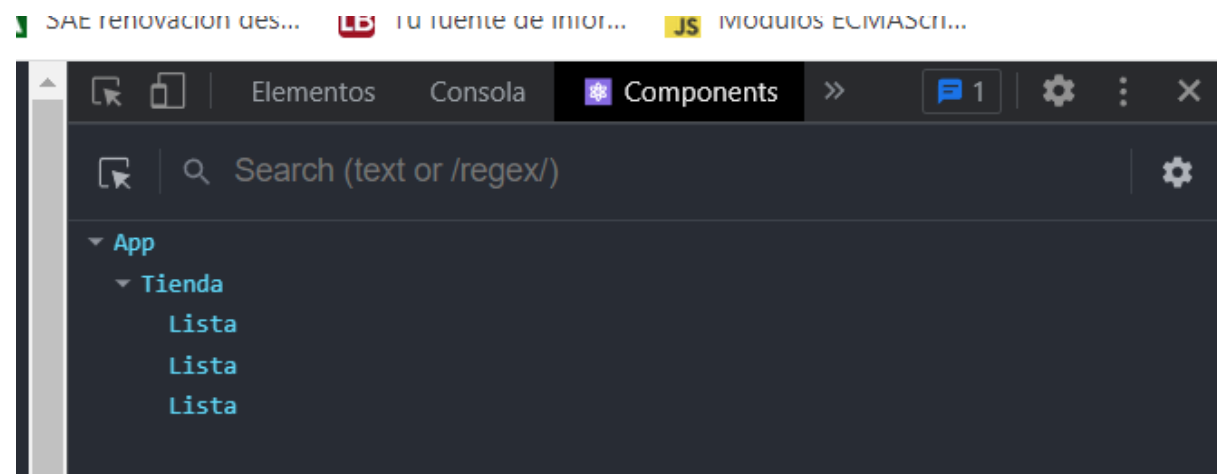
Y creamos la función del componente:

```
24   export default function Tienda(){  
25     return(  
26       <>  
27         <Lista productos = {electronica} categoria="Electronica"/>  
28         <Lista productos = {alimentacion} categoria="Alimentacion"/>  
29         <Lista productos = {mascotas} categoria="Mascotas"/>  
30       </>  
31     )  
32   }
```

Y en nuestro archivo App.jsx sólo tenemos que importar y llamar al componente Tienda:

```
3  import Tienda from './components/Tienda/Tienda';
4
5  function App() {
6    return (
7      <div className="App">
8        <header className="App-header">
9          <p>Ejercicio 2 - React</p>
10         <p>Abel Rios</p>
11         <Tienda/>
12       </header>
13     </div>
14   );
15 }
```

Si entramos al navegador, con la extensión que instalamos ayer, podemos ver los componentes:



### \*\*\*Desestructuración\*\*\*

Podemos “dividir” el objeto, o mejor dicho, sacar las propiedades a una copia fuera para mejor manejo. Para ello, por ejemplo:

```
const usuario = {  
  name: "Nacho",  
  surname: "Viano",  
  age: 23,  
  phone: 66666666  
};
```

```
const { name, surname, ...rest } = usuario;
```

Estamos creando una constante “name”, una constante “surname”, una constante “rest” con el resto de propiedades. Si las propiedades son simples (string, número, boolean) se pasan por valor, pero si la propiedad es un objeto, se pasa por referencia y si se modifica, se modifica en el objeto primario:

Código:

```
1  const usuario = {  
2    name: "Nacho",  
3    surname: "Viano",  
4    age: 23,  
5    phone: 66666666,  
6    address: {  
7      street: "Calle larios",  
8      number: 3  
9    }  
10 };  
  
11  
12 const array = ["Nacho", "Raul", "Rocio", "Paula"];  
13  
14 let { name, surname, address } = usuario;  
15 address.street = "Pepito";  
16 console.log(usuario);  
17  
18
```

Consola:

```
▼ {name: "Nacho", surname: "Viano", age: 23, phone: 66666666, address: Object}  
  name: "Nacho"  
  surname: "Viano"  
  age: 23  
  phone: 66666666  
  ▼ address: Object  
    street: "Pepito"  
    number: 3
```

Esto mismo se puede hacer con arrays también, y metemos el valor de la posición del array en una variable:

```
1  const array = ["Nacho", "Raul", "Rocio", "Paula"];  
2  const [profe, ...rest] = array  
3  
4  console.log(profe)
```

consola:

**Nacho**