

# LISTAS - Renderizado de multiples componentes

- React nos permite utilizar un array transformarlo en una lista de elementos,es decir que por medio de una lista de datos podemos iterar sobre la misma y crear un componente o elemento por cada uno de ellos.
- Esto es posible haciendo uso de un metodo de JS : map(metodo que crea un nuevo array con los resultados de la llamada a la función indicada aplicados a cada uno de sus elementos.)
- Lo que nos permite este metodo es iterar sobre los items del array y dentro de la funcion que le podemos asignar a map, crear un elemento o componente por cada uno de ellos e incorporar su contenido a estos.
- Por ejemplo ,podemos tener una lista de numeros y generar un <li> por cada uno de ellos, que contengan dicho numero.
- Para crear dichos elementos podemos hacerlo tanto en una variable que luego incorporamos al return de un componente, como directamente dentro del return de un componente ,utilizando llaves donde colocamos esta iteracion.
- Una vez ejecutado esto,nos mostrara el resultado al renderizarse,pero React nos va a tirar una advertencia sobre algo llamado *key* que vamos a aprender a continuacion..

## KEY PROPS

- Cuando queremos renderizar una lista , React almacena información acerca de cada elemento de la lista renderizada.Esto le permite que cuando tengamos un cambio en alguna de ellas (como puede ser una modificacion en su estructura,contenido o incluso eliminar una de ellas ),pueda tener una idea de cual de ellas fue modificada y pueda hacer la actualizacion en el DOM de una manera mas eficiente.
- React es un programa de computadora y no puede saber nuestras intenciones, por lo que necesitamos especificar una propiedad llamada **key** para cada elemento de la lista para diferenciar cada uno de sus hermanos.
- Las **keys** ayudan a React a identificar que ítems han cambiado, son agregados, o son eliminados.
- **key** es una propiedad especial y reservada en React que lleva como valor un string, que almacena directamente en el elemento cuando este es creado.
- Aunque key pertenece a las props, no puede referenciarse en el objeto de las props ,es decir que un componente no puede averiguar sobre su key,solo es un dato que utiliza React para poder ubicar de una mejor manera que item de la lista se modifiko
- Si no la colocamos,React emitira una advertencia y resolvera el problema del cambio en el item de la manera que concidere.
- Al asignar un nombre a la **key**, no se necesita que sea globalmente unico,solo que sea unico entre sus componentes y hermanos.

## Ejemplo

### Codigo

```
JS App.js M X
src > JS App.js > ...
You, 4 seconds ago | 1 author (You)
1 import Filtros from './Filtros';
2 import './App.css';
3
4 export default function App() {
5   let numeros=[1,2,3,4,5]
6   return (
7     <div className="App">
8       <Filtros dato={numeros}/>
9     </div>
10  );
11 }
```

```
JS Filtros.js U X
src > JS Filtros.js > Filtros
1 export default function Filtros({dato}){
2
3   return(
4     <div>
5       <h2>HOLA</h2>
6       {dato.map((el)=> <p key={el}>{el}</p>)}
7     </div>
8   )
9 }
```

### Navegador

