

## Nivel 2 React - 14/01/2026 - Iker Ríos Navarro

Primero prueba con un <div> envolviendo todo:

The screenshot shows a browser window at `localhost:5173` displaying the text "Nivel 1 de React desbloqueado" and "Hola Klara, como estas?". To the right is a code editor with `App.jsx` open, containing the following code:

```
1 export default function App() {
2   return (
3     <div>
4       <h1>Nivel 1 de React desbloqueado</h1>
5       <h2>Hola Klara, como estas?</h2>
6     </div>
7   )
8 }
```

Luego cambia el <div> por un Fragment vacío: <> ... </> y guarda:

The screenshot shows the same browser and code editor. The browser still displays the same text. The code editor now shows the following code:

```
1 export default function App() {
2   return (
3     <>
4       <h1>Nivel 1 de React desbloqueado</h1>
5       <h2>Hola Klara, como estas?</h2>
6     </>
7   )
8 }
```

¿Qué es `getElementById('root')`?:

`document.getElementById('root')` obtiene del HTML el elemento con id root, que es el contenedor principal de la app. React usa ese elemento como punto de entrada para montar y renderizar toda la interfaz dentro de él.

Comentarios:

The screenshot shows a code editor with `App.jsx` open. A multi-line comment block is present between lines 11 and 18, starting with `/*` and ending with `*/`. The code before the comment is:

```
1 export default function App() {
2   return (
3     <>
4       <h1>Nivel 1 de React desbloqueado</h1>
5       <h2>Hola Klara, como estas?</h2>
6     </>
7   )
8 }
```

The code after the comment is:

```
11 /*
12  * comentario
13  * pedido
14  * por
15  * Klara
16  */
17
18 }
```

**Crea 3 componentes sencillos (UserComponent, ProfileComponent, FeedComponent):**

The screenshot shows a browser window at `localhost:5173` displaying three simple components: **User component**, **Profile component**, and **Feed component**. Below the browser is a code editor showing the `App.jsx` file. The code defines a `ParentComponent` which returns a single `<div>` element containing the three components.

```
src > App.jsx > ...
1  export default function App() {
2    return <ParentComponent />
3  }
4
5  function ParentComponent() {
6    return (
7      <div>
8        <UserComponent />
9        <ProfileComponent />
10       <FeedComponent />
11     </div>
12   )
13 }
14
15 function UserComponent() {
16   return <h2>User component</h2>
17 }
18
19 function ProfileComponent() {
20   return <h2>Profile component</h2>
21 }
22
23 function FeedComponent() {
24   return <h2>Feed component</h2>
25 }
26
27 !
```

### 1. ¿Qué es un componente en React?

Es una función (o clase) que devuelve JSX y representa una parte de la interfaz, como un botón, un perfil o una pantalla.

### 2. ¿Por qué usamos Fragmentos (`<>...</>`) en lugar de un `<div>`?

Para agrupar varios elementos sin añadir nodos extra al HTML, evitando `<div>` innecesarios en el DOM.

### 3. ¿Qué papel tienen `index.html` y `main.jsx` en el renderizado?

`index.html` contiene el `<div id="root">` y `main.jsx` le dice a React qué componente (App) debe renderizar dentro de ese root.

### 4. ¿Qué significa “componer componentes”?

Es construir componentes grandes usando otros más pequeños, combinándolos como piezas para formar toda la interfaz.