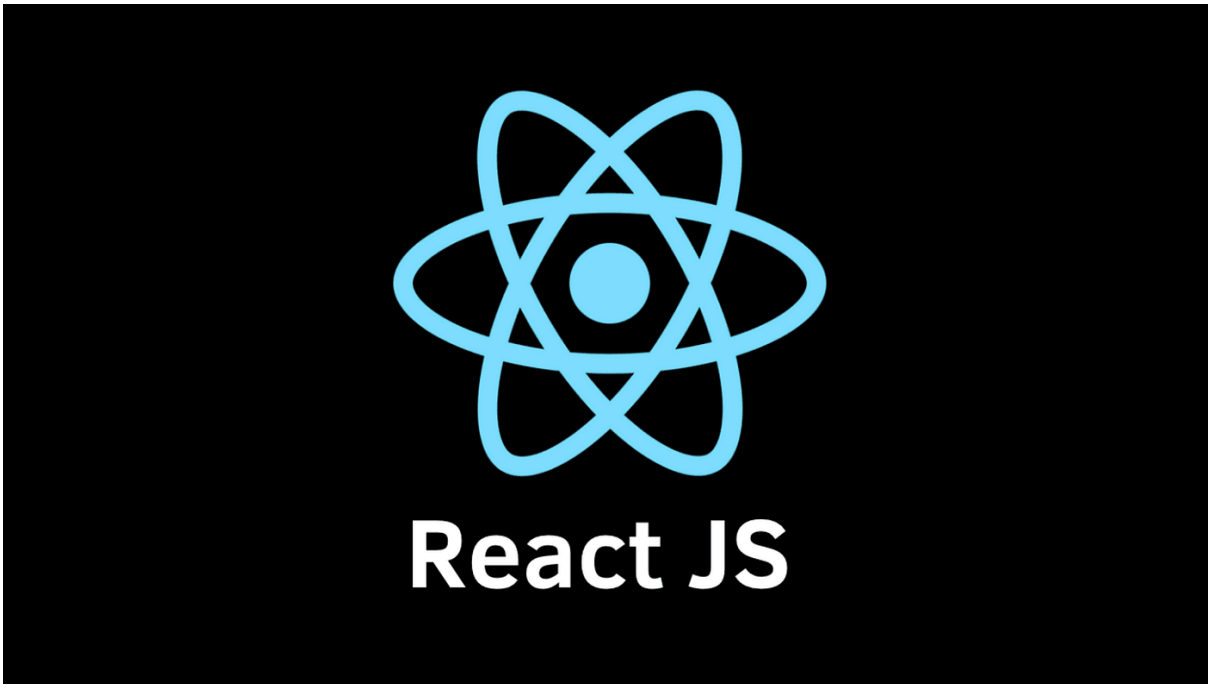
Desarrollo web en entorno cliente - Nivel 2 React

Pablo Valero López *14/01/2026*

2ºDAW



Índice:

Objetivos

- *Entender qué es un componente en React y cómo se define.*
- *Aprender a devolver uno o varios elementos usando Fragmentos (<>...</>).*
- *Saber cómo se renderiza React en el navegador (root + ReactDOM).*
- *Escribir comentarios dentro y fuera del JSX sin romper el render.*
- *Componer varios componentes para construir una interfaz en forma de árbol.*

1) Ejecuta el proyecto

1. Abre una terminal en la carpeta my-react-app.
2. Ejecuta: npm install (si no lo hiciste).
3. Inicia el servidor: npm run dev
4. Abre la URL local que te muestre la terminal (por ejemplo: http://localhost:5173/).

En la siguiente imagen se muestra el proyecto iniciado.

2) Devuelve múltiples elementos con Fragment

5. Abre src/App.jsx y reemplaza el contenido por el ejemplo con dos encabezados.
6. Primero prueba con un envolviendo todo.
7. Luego cambia el por un Fragment vacío: <> … </> y guarda.

En la siguiente imagen se muestra el Fragment vacío con el h1 y el comentario en su interior.

```jsx
export default function App() {
return (
        <>
        <h1>Segundo nivel de React</h1>
        {/*Comentario dentro de JSX*/}
        <Componentes/>
        </>
    )
}
```

3) Renderizado: ¿dónde se “engancha” React?

8. Abre index.html y localiza: .
9. Abre src/main.jsx y localiza createRoot(...).render(...).
10. Escribe en tu informe (2-3 líneas) qué hace document.getElementById('root').

En la siguiente imagen se muestra getElementById(‘root’).

```jsx
ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
)
```

*document.getElementById('root') sirve como el **punto de anclaje** o contenedor vacío donde React "inyectará" toda la aplicación para que sea visible en el navegador.*

4) Comentarios en React

11. Añade un comentario con // fuera del return.

12. Luego comenta una línea dentro del JSX usando: {/ ... */}.*

13. Usa el atajo de VSCode: Ctrl+/ (Windows/Linux) o Cmd+/ (macOS).

En la siguiente imagen se muestran los comentarios fuera y dentro de JSX.

App.jsx M main.jsx

Desarrollo_Web_Entorno_Cliente > nivel1_react > my-react-app

```jsx
export default function App() {
return (
        <>
        <h1>Segundo nivel de React</h1>
        {/*Comentario dentro de JSX*/}
        <Componentes/>
        </>
    )
}

function Componentes(){
    return (
        <>
        <UserComponent/>
        <ProfileComponent/>
        <FeedComponent/>
        </>
    )
}

function UserComponent(){
    return <>
        <h2>1º Componente</h2>
        </>
}
function ProfileComponent(){
    return <>
        <h3>2º Componente</h3>
        </>
}
function FeedComponent(){
    return <>
        <h4>3º Componente</h4>
        </>
}
// Comentario fuera de return
// export default App
```

5) Composición: varios componentes dentro de uno

14. Crea 3 componentes sencillos (UserComponent, ProfileComponent, FeedComponent).

15. Crea un ParentComponent que los renderice en orden.

16. Haz que App renderice <ParentComponent />.

En la siguiente imagen se muestra los componentes.

App.jsx M main.jsx

Desarrollo_Web_Entorno_Cliente > nivel1_react > my-react-app

```jsx
export default function App() {
return (
        <>
        <h1>Segundo nivel de React</h1>
        {/*Comentario dentro de JSX*/}
        <Componentes/>
        </>
    )
}

function Componentes(){
    return (
        <>
        <UserComponent/>
        <ProfileComponent/>
        <FeedComponent/>
        </>
    )
}

function UserComponent(){
    return <>
        <h2>1º Componente</h2>
        </>
}
function ProfileComponent(){
    return <>
        <h3>2º Componente</h3>
        </>
}
function FeedComponent(){
    return <>
        <h4>3º Componente</h4>
        </>
}
// Comentario fuera de return
// export default App
```

Preguntas extra para el PDF

1. ¿Qué es un componente en React?

Un componente en React es una pieza de código ***independiente y reutilizable*** *que se encarga de mostrar una parte específica de la interfaz de usuario.*

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

Usamos Fragmentos principalmente para ***evitar "ensuciar" el HTML final*** *con elementos innecesarios. En React, un componente* ***siempre debe devolver un solo elemento padre****. Si tienes dos elementos y no quieres encerrarlos en un* `<div>`*, usas un Fragmento.*

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

El ***`index.html`*** *proporciona la* ***"caja vacía" (`id="root"`)*** *y carga el script, mientras que* ***`main.jsx`*** *busca esa caja e* ***inyecta tu aplicación (`<App />`)*** *dentro de ella para que sea visible.*

4. Explica con tus palabras qué significa "componer componentes".

Significa construir interfaces complejas insertando componentes pequeños y específicos dentro de componentes más grandes.