

---

# Nivel 3 - JSX y Listas en React

---

## Tabla de contenido

1. Configuración y Ejecución del Proyecto .....	1
2. Desarrollo de la Actividad.....	1
3. Preguntas Extra.....	3
4. Código CSS Aplicado (App.css).....	4

## 1. Configuración y Ejecución del Proyecto

He inicializado bien el proyecto , verificado la instalación de dependencias mediante npm install y arrancando el servidor de desarrollo.

```
PS C:\Users\Ibra\Desktop\2º DAW\Desarrollo web en entorno cliente\Entorno-Cliente\Entorno-Cliente\2º trimestre\Tema 5\React\nivel2_react\my-react-app> npm run dev

> my-react-app@0.0.0 dev
> vite

✓ Console Ninja extension is connected to Vite, see https://tinyurl.com/2vt8jxzw
Re-optimizing dependencies because vite config has changed

VITE v5.4.21 ready in 1693 ms

→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
15:51:14 [vite] hmr update /src/App.jsx
15:51:14 [vite] hmr update /src/App.jsx (x2)
```

---

## 2. Desarrollo de la Actividad

### Parte A y B: Uso de JSX y Expresiones

He sustituido el contenido de App.jsx para trabajar con elementos JSX almacenados en variables. Ademas he utilizado las llaves {} para renderizar contenido dinámico y aplicar clases de CSS mediante el atributo className.

### Parte C y D: Renderizado de Listas y Estilos Dinámicos

He implementado una lista de usuarios utilizando el método `.map()`. Siguiendo las instrucciones, añadiéndome a mi mismo (Ibrahim) y se ha aplicado estilos CSS diferenciados según el rol de cada usuario para mejorar la interfaz.

```
src > App.jsx > App > users
You, 10 minutes ago | 1 author (You)
1 import './App.css'
2
3 function App() {
4 // Parte A: Guardar JSX en una variable
5 const bienvenida = <h1>Bienvenido al multiverso JSX de Klara</h1>;
6
7 // Parte B: Expresiones con { }
8 const lowercaseClass = 'text-lowercase';
9 const instructionText = 'Hola, humano esclavizado de DAW2. React detectado check';
10
11 // Parte C: Lista con map() y nuevos usuarios
12 const users = []
13   { id: 1, name: 'Jose', role: 'Web Developer' },
14   { id: 2, name: 'Estefanía', role: 'Web Designer' },
15   { id: 3, name: 'Rubén', role: 'Team Leader' },
16   { id: 4, name: 'Ibrahim', role: 'Full Stack' }, // me incluyo 🤗 You, 2
17 ];
18
19 return (
20   <div className="contenedor">
21     {/* Renderizamos la variable de la Parte A */}
22     {bienvenida}
23
24     {/* Renderizamos expresiones de la Parte B */}
25     <h2 className={lowercaseClass}>{instructionText}</h2>
26
27     <hr />
28
29     <p>Lista de usuarios activos:</p>
30     <ul className="user-list">
31       {users.map((user) => {
32         // Parte D: Aplicar estilos rápidos basados en el rol
33         const roles = user.role === 'Team Leader' ? 'leader' : 'member';
34
35         return (
36           <li key={user.id} className={roles}>
37             <strong>{user.name}</strong> - {user.role}
38           </li>
39         );
40       })}
41     </ul>
42   </div>
43 );
44 }
45
46 export default App;
```



### 3. Preguntas Extra

#### 1. ¿Qué es JSX con tus palabras?

Es una forma de escribir código que se parece a HTML pero dentro de JavaScript. React lo usa para que sea mucho más fácil e intuitivo diseñar cómo se va a ver la página web.

#### 2. ¿Por qué usamos las llaves dentro de JSX? Pon un ejemplo.

Las llaves nos permiten indicarle a React que lo que viene a continuación no es texto, sino código JavaScript. Esto es útil para incluir variables o realizar cálculos dentro del código JSX. Por ejemplo, si queremos mostrar un saludo personalizado, podemos hacerlo de esta manera: <h1>Hola, {nombre}</h1>. En este caso, el nombre se muestra como una variable dentro de las llaves, lo que nos permite cambiar el texto según sea necesario.

#### 3. ¿Para qué sirve la prop key en listas?

Es como un DNI para cada elemento de la lista. Ayuda a React a saber exactamente qué elemento ha cambiado o se ha borrado, lo que hace que la web cargue mucho más rápido.

#### **4. ¿Por qué usamos className y no class en React?**

Porque en JavaScript la palabra class ya se usa para crear clases de objetos. Para no confundir al ordenador, React decidió usar className para las clases de CSS.

---

#### [4. Código CSS Aplicado \(App.css\)](#)

Para diferenciar los roles, se han definido las siguientes clases:

##### **CSS**

```
.text-lowercase { text-transform: lowercase; color: #3498db; }

.leader { background-color: #f1c40f; padding: 10px; border-radius: 8px; margin: 5px 0; }

.member { background-color: #ecf0f1; padding: 10px; border-radius: 8px; margin: 5p
```