

Nivel 3 React

DAW2-DWEC

Índice

Parte A: JSX básico (guardar y devolver un elemento).....	2
Parte B: Expresiones dentro de JSX con { }.....	2
Parte C: Renderizar una lista con map() + key.....	3
Parte D: className y estilos rápidos.....	5
Preguntas extra para el PDF.....	7

Parte A: JSX básico (guardar y devolver un elemento)



“Figura 01, jsx en una variable, elaboración propia”

Parte B: Expresiones dentro de JSX con { }



“Figura 02, expresiones dentro de JSX, elaboración propia”

The screenshot shows a code editor with a dark theme. The top navigation bar includes 'final' and 'Help' on the left, and a search bar on the right containing 'nivel3_react'. Below the navigation is a tab bar with three tabs: 'App.jsx' (selected), 'main.jsx', and '# index.css'. The main workspace displays the following code:

```
my-react-app > src > main.jsx
1 import React from 'react'
2 import ReactDOM from 'react-dom/client'
3 import App from './App.jsx'
4 import './index.css' // Line 4 is highlighted with a red box
5
6 ReactDOM.createRoot(document.getElementById('root')).render(
7   <React.StrictMode>
8     <App />
9   </React.StrictMode>,
10 )
11
```

“Figura 03, había que incluir la línea que importa los estilos del index.css, elaboración propia”

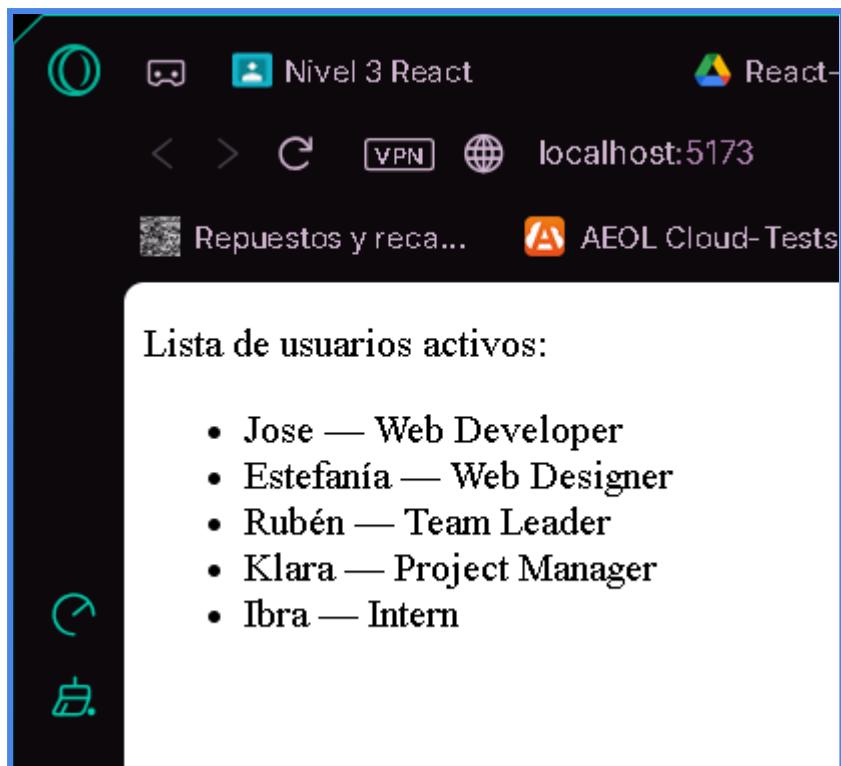
The screenshot shows the VS Code interface with the following details:

- Terminal** tab is selected.
- Help** tab is visible.
- File Explorer** sidebar shows files: `App.jsx`, `main.jsx`, and `index.css`.
- Content Area**:
 - File path: `my-react-app > src > # index.css > .text-lowercase`
 - Code content:

```
1 .text-lowercase {  
2   text-transform: lowercase;  
3 }
```

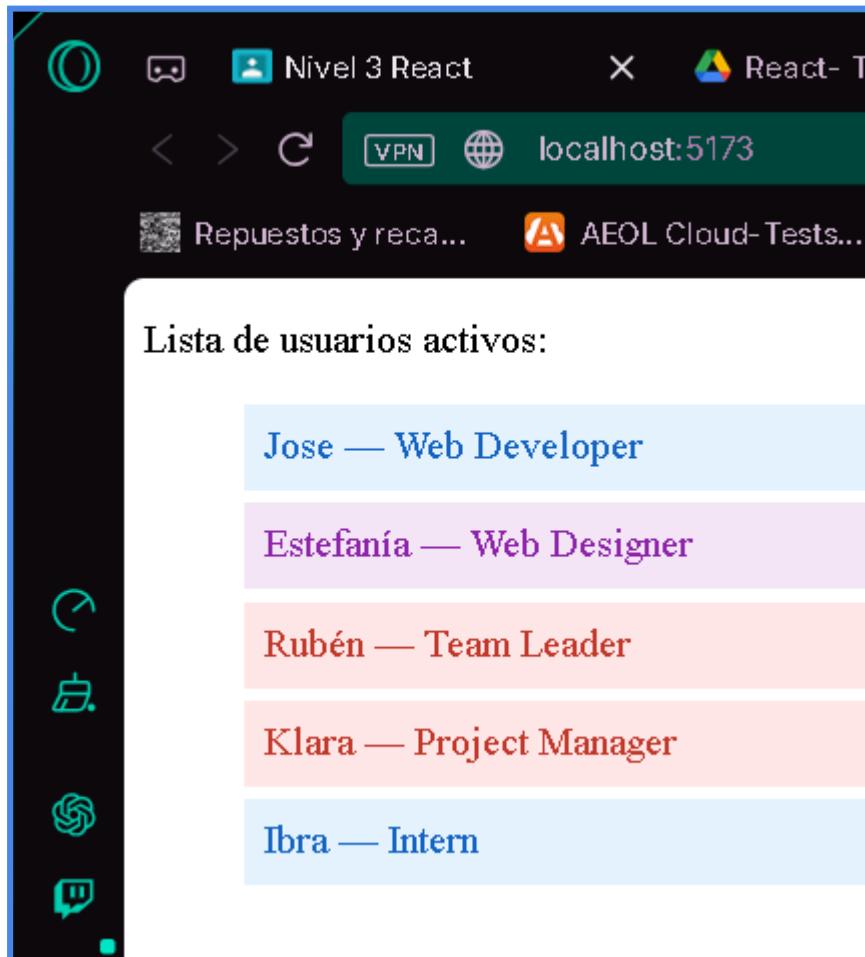
“Figura 04, index.css, elaboración propia”

Parte C: Renderizar una lista con map() + key



“Figura 05, lista renderizada, elaboración propia”

Parte D: className y estilos rápidos



“Figura 06, className para diferenciar roles, elaboración propia”

The screenshot shows a code editor interface with a dark theme. At the top, there are tabs for "App.jsx", "# index.css", and "main.jsx". The "# index.css" tab is currently active. Below the tabs, the file path "my-react-app > src > # index.css > .role-boss" is displayed. The code itself is a CSS file containing the following styles:

```
1  .text-lowercase {  
2  |   |   text-transform: lowercase;  
3  | }  
4  
5  li {  
6  |   padding: 8px;  
7  |   margin-bottom: 5px;  
8  |   list-style: none;  
9  | }  
10  
11 .role-boss {  
12 |   background-color: #ffe6e6;  
13 |   color: #c0392b;  
14 | }  
15  
16 .role-design {  
17 |   background-color: #f3e5f5;  
18 |   color: #8e24aa;  
19 | }  
20  
21 .role-tech {  
22 |   background-color: #e3f2fd;  
23 |   color: #1565c0;  
24 | }
```

"Figura 07, css de los roles, elaboración propia"

Preguntas extra para el PDF

1. ¿Qué es JSX con tus palabras?

- JSX es una extensión de la sintaxis de JavaScript que permite escribir la estructura de la interfaz de usuario de una manera muy similar al HTML

2. ¿Por qué usamos {} dentro de JSX? Pon un ejemplo.

- Las llaves se usan en JSX para salir del contexto de "etiquetado visual" y entrar en el contexto de JavaScript. Se usan como una ventana que nos permite insertar valores dinámicos, variables o expresiones lógicas directamente en la estructura.
- Por ejemplo, si tenemos una variable definida como const edad = 25, al escribir <p>Tienes {edad} años</p>, el navegador mostrará 25. Si no se usan las llaves, el navegador mostraría literalmente la palabra "edad".

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

- La propiedad key es un mecanismo importante para el rendimiento de React. Sirve para darle una identidad única y estable a cada elemento dentro de una lista que se genera de forma dinámica.

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

- El motivo es que JSX convive dentro de JavaScript, y en este lenguaje la palabra class es una palabra reservada que se utiliza para definir clases en la programación orientada a objetos.