

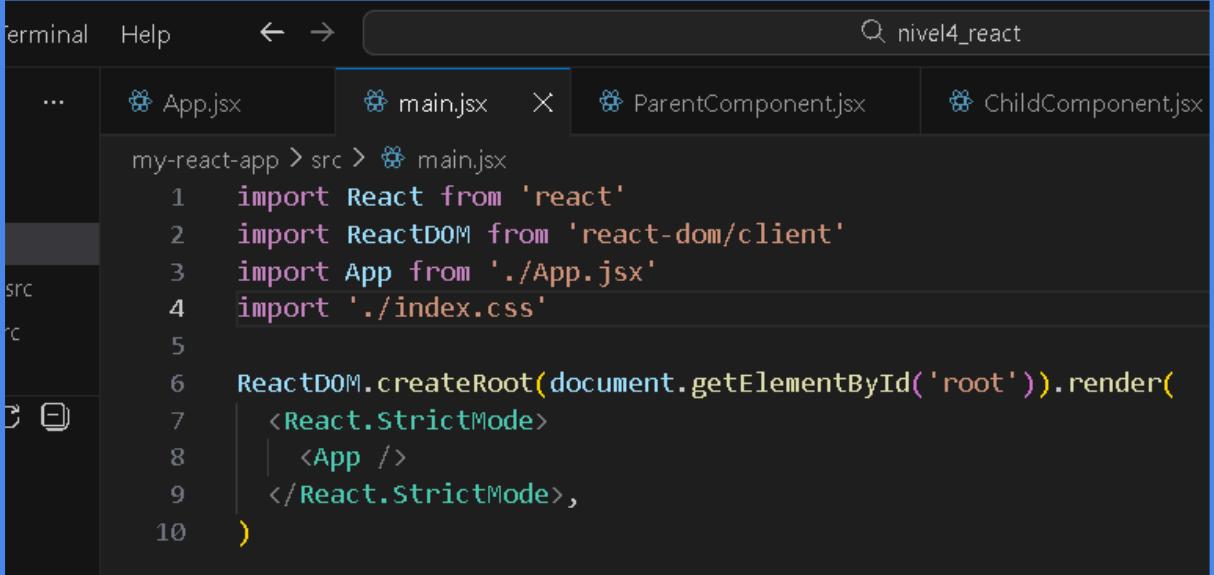
Nivel 4 React

DAW2-DWEC

Índice

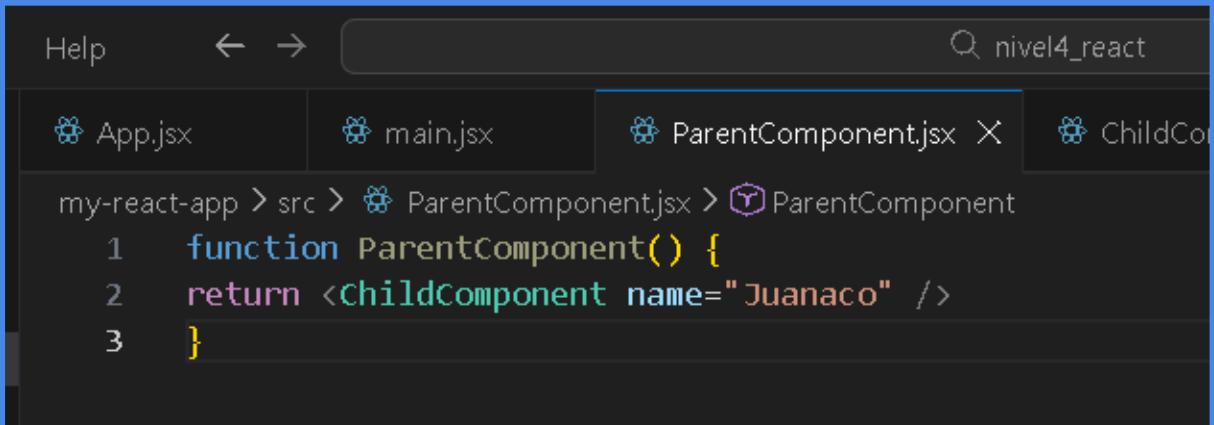
Parte A: Props básicas (Padre -> Hijo).....	2
Parte B: Pasar múltiples props.....	3
Parte C: Pasar funciones como props.....	5
Parte D: Las props son inmutables.....	6
Parte E: Estado con useState.....	8
Parte F: Pasar estado y setState a un componente hijo.....	10
Parte G: Inspeccionar props y state con React DevTools.....	12

Parte A: Props básicas (Padre -> Hijo)



```
Terminal Help ← → Q nivel4_react
...
App.jsx main.jsx X ParentComponent.jsx ChildComponent.jsx
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'
5
6 ReactDOM.createRoot(document.getElementById('root')).render(
7   <React.StrictMode>
8     <App />
9   </React.StrictMode>,
10 )
```

“Figura 01, main.jsx, elaboración propia”



```
Help ← → Q nivel4_react
...
App.jsx main.jsx ParentComponent.jsx X ChildComponent.jsx
my-react-app > src > ParentComponent.jsx > ParentComponent
1 function ParentComponent() {
2   return <ChildComponent name="Juanaco" />
3 }
```

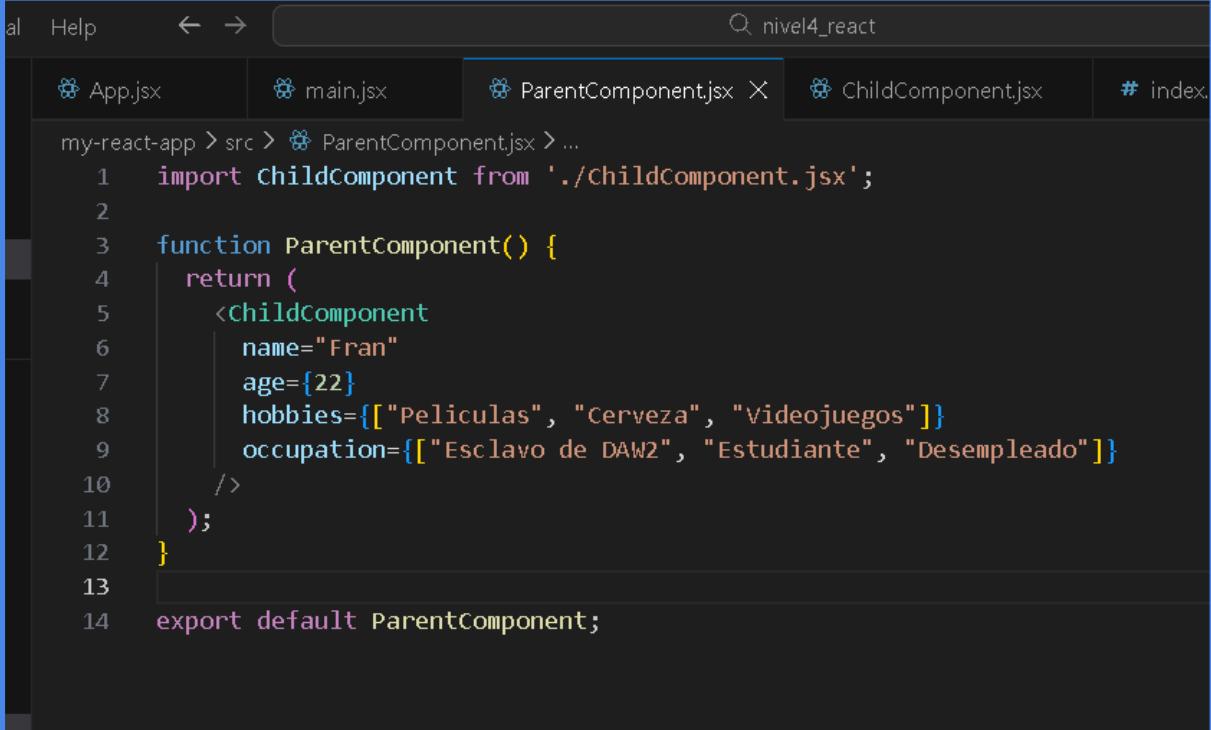
“Figura 02, componente padre, elaboración propia”



```
al Help ← → Q nivel4_react
...
App.jsx main.jsx ParentComponent.jsx ChildComponent.jsx
my-react-app > src > ChildComponent.jsx > ChildComponent
1 function ChildComponent(props) {
2   return <p>Hello bros! my name is {props.name}</p>
3 }
```

“Figura 03, componente hijo, elaboración propia”

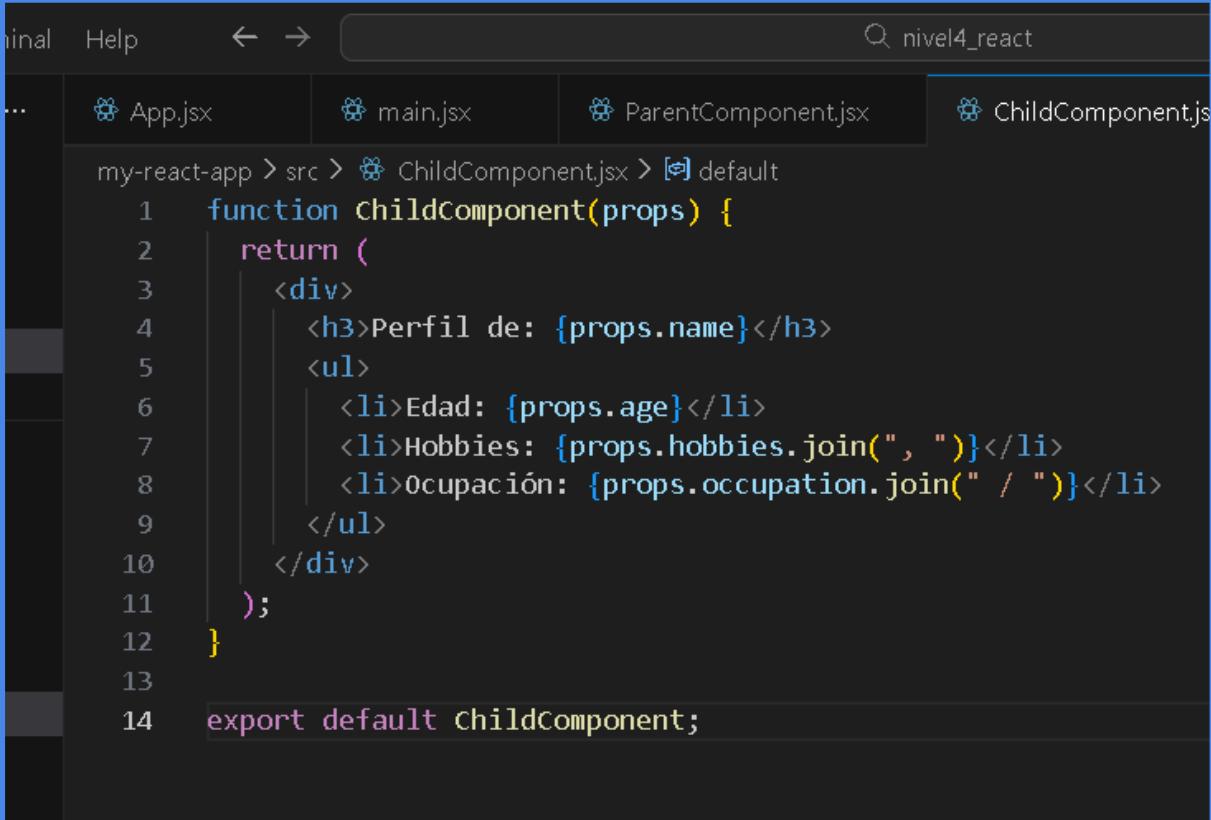
Parte B: Pasar múltiples props



The screenshot shows a code editor with the file `ParentComponent.jsx` open. The code defines a component that imports `ChildComponent` and returns it with various props. The props include `name`, `age`, `hobbies` (an array of strings), and `occupation` (another array of strings). The code is part of a project named `my-react-app`.

```
my-react-app > src > ParentComponent.jsx > ...
1 import ChildComponent from './ChildComponent.jsx';
2
3 function ParentComponent() {
4     return (
5         <ChildComponent
6             name="Fran"
7             age={22}
8             hobbies={['Peliculas", "Cerveza", "Videojuegos']}
9             occupation={['Esclavo de DAW2", "Estudiante", "Desempleado']}
10        />
11    );
12 }
13
14 export default ParentComponent;
```

“Figura 04, Componente padre con el hijo referenciado, elaboración propia”



The screenshot shows a code editor with the file `ChildComponent.jsx` open. The code defines a component that takes `props` and returns a `div` containing an `h3` for the name and a `ul` for the hobbies and occupation, separated by commas. The code is part of a project named `my-react-app`.

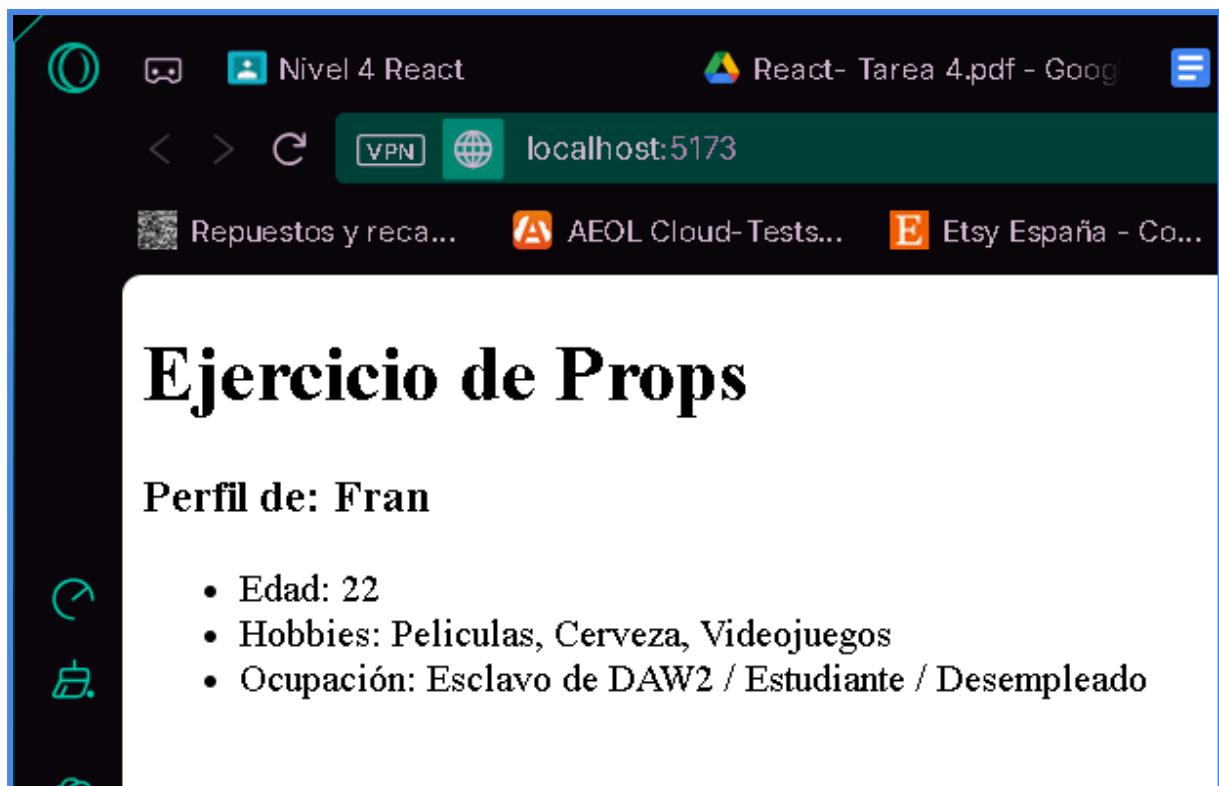
```
my-react-app > src > ChildComponent.jsx > default
1 function ChildComponent(props) {
2     return (
3         <div>
4             <h3>Perfil de: {props.name}</h3>
5             <ul>
6                 <li>Edad: {props.age}</li>
7                 <li>Hobbies: {props.hobbies.join(", ")})</li>
8                 <li>Ocupación: {props.occupation.join(" / ")})</li>
9             </ul>
10        </div>
11    );
12 }
13
14 export default childComponent;
```

“Figura 05, Componente hijo con toda la info, elaboración propia”

The screenshot shows a code editor window with the title bar "Terminal Help" and a search bar "n nivel4_react". The left sidebar shows a tree structure with "app\src" and "app\src". The main area displays the content of the "App.jsx" file:

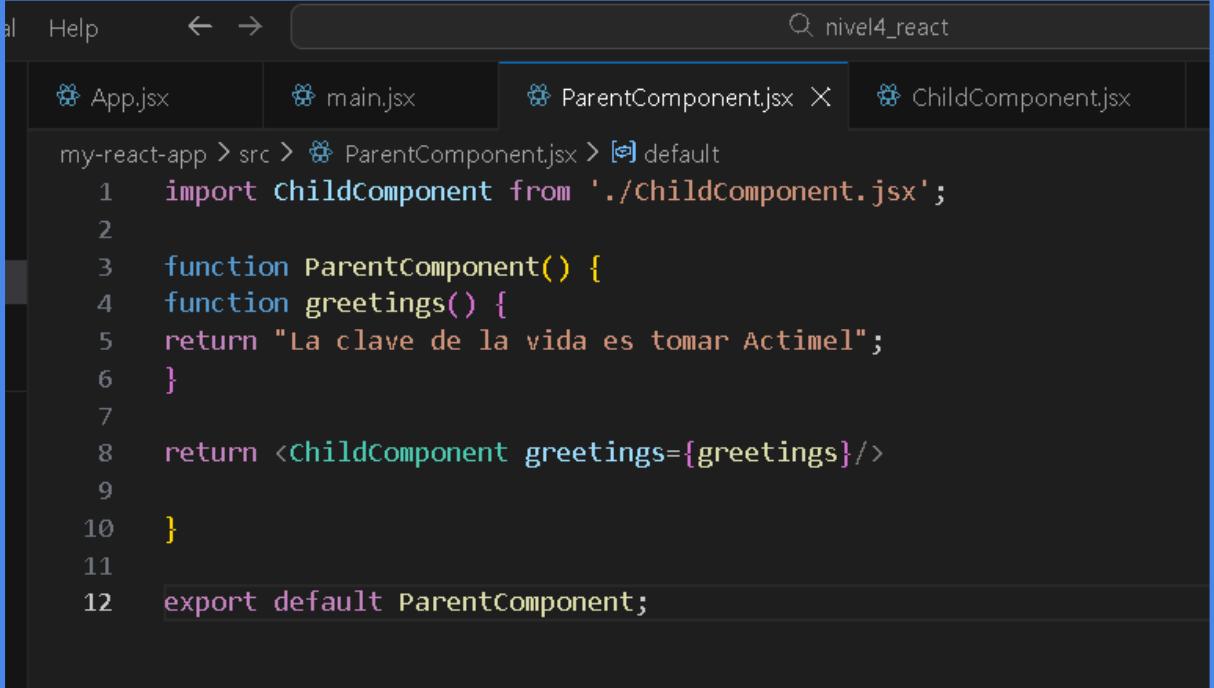
```
1 import ParentComponent from './ParentComponent.jsx';
2
3 function App() {
4     return (
5         <div>
6             <h1>Ejercicio de Props</h1>
7             <ParentComponent />
8         </div>
9     );
10 }
11
12 export default App;
```

“Figura 06, app.jsx, elaboración propia”



“Figura 07, muestra en el navegador, elaboración propia”

Parte C: Pasar funciones como props



The screenshot shows a code editor window with the title bar "nivel4_react". The tabs at the top are "App.jsx", "main.jsx", "ParentComponent.jsx", and "ChildComponent.jsx". The "ParentComponent.jsx" tab is active. The code in the editor is:

```
my-react-app > src > ParentComponent.jsx > default
1 import ChildComponent from './ChildComponent.jsx';
2
3 function ParentComponent() {
4   function greetings() {
5     return "La clave de la vida es tomar Actimel";
6   }
7
8   return <ChildComponent greetings={greetings}/>
9
10 }
11
12 export default ParentComponent;
```

“Figura 08, Componente padre, elaboración propia”



The screenshot shows a code editor window with the title bar "nivel4_react". The tabs at the top are "App.jsx", "main.jsx", and "ParentComponent.jsx" (which is faded). The active tab is "ChildComponent.jsx". The code in the editor is:

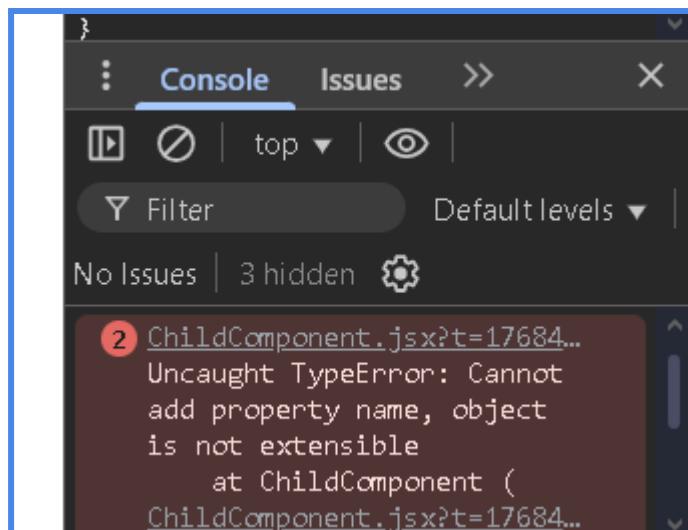
```
my-react-app > src > ChildComponent.jsx > default
1 function ChildComponent(props) {
2   return <p>{props.greetings()}</p>
3 }
4
5 export default ChildComponent;
```

“Figura 09, Componente hijo, elaboración propia”



“Figura 10, muestra en el navegador, elaboración propia”

Parte D: Las props son inmutables



“Figura 11, el error no se me muestra en la pantalla asi que he tenido que acceder a la consola del inspeccionar, elaboración propia”

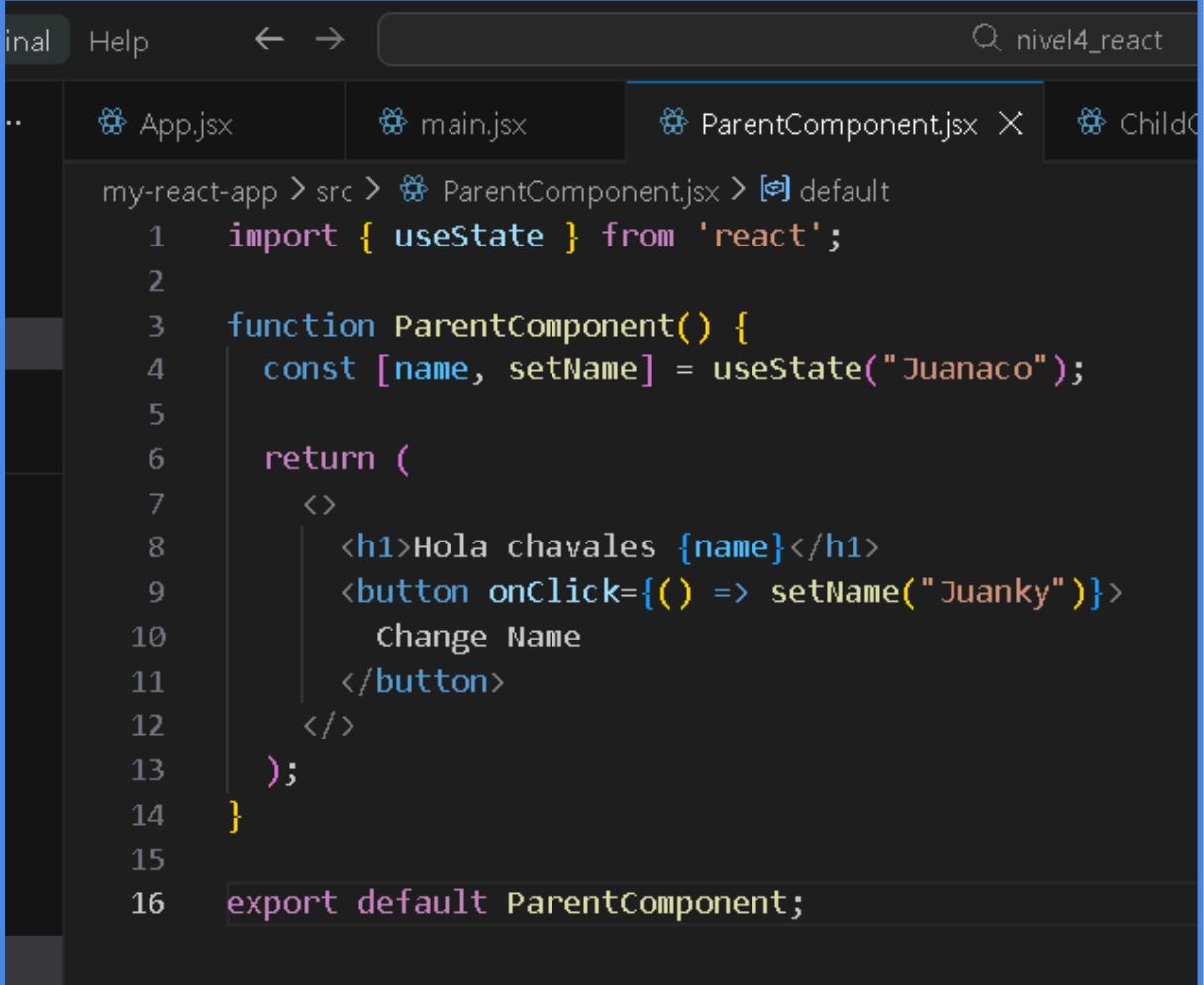


The screenshot shows a code editor interface with a dark theme. At the top, there's a navigation bar with 'Terminal' and 'Help' on the left, and a search bar on the right containing the text 'nivel4_react'. Below the navigation bar, there are tabs for 'App.jsx', 'main.jsx', 'ParentComponent.jsx', and 'ChildComponent.jsx', with 'ChildComponent.jsx' being the active tab. The code editor displays the following code:

```
my-react-app > src > ChildComponent.jsx > ChildComponent
1 function ChildComponent(props) {
2   props.name = "Pablo";
3
4   return <p>Hello profesora! my name is {props.name}</p>;
5 }
6 export default ChildComponent;
```

“Figura 12, Componente hijo, elaboración propia”

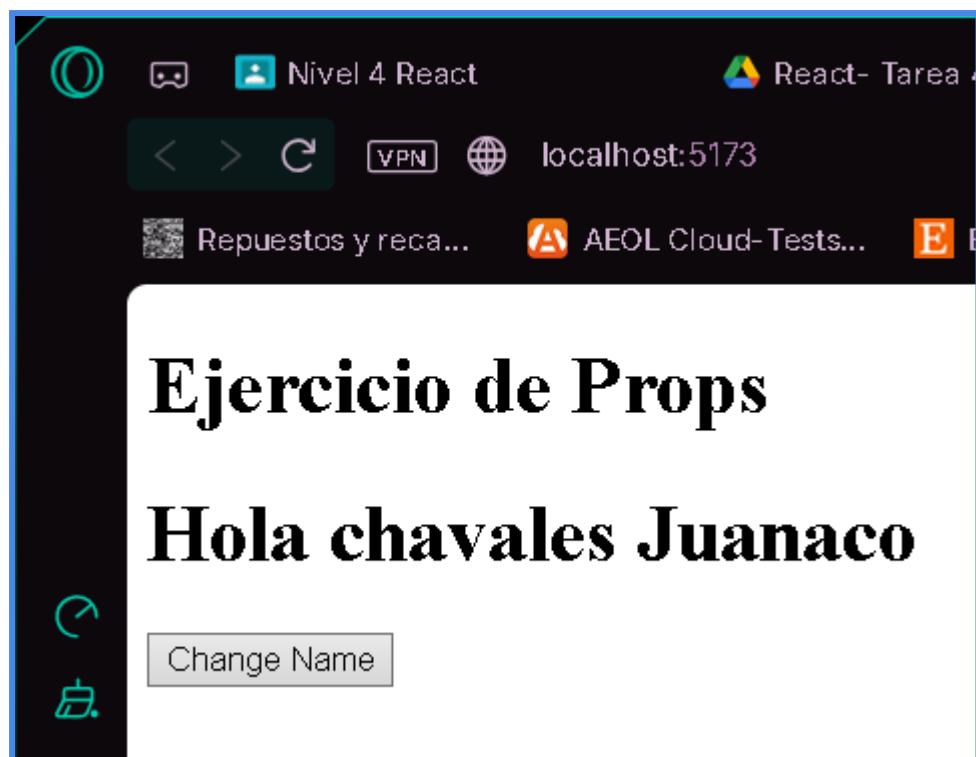
Parte E: Estado con useState



The screenshot shows a code editor interface with a dark theme. At the top, there are tabs for 'inal' (disabled), 'Help', and several files: 'App.jsx', 'main.jsx', 'ParentComponent.jsx' (which is the active tab), and 'ChildC'. A search bar at the top right contains the text 'nivel4_react'. Below the tabs, a breadcrumb navigation shows the file path: 'my-react-app > src > ParentComponent.jsx > default'. The main area displays the code for the 'ParentComponent' function:

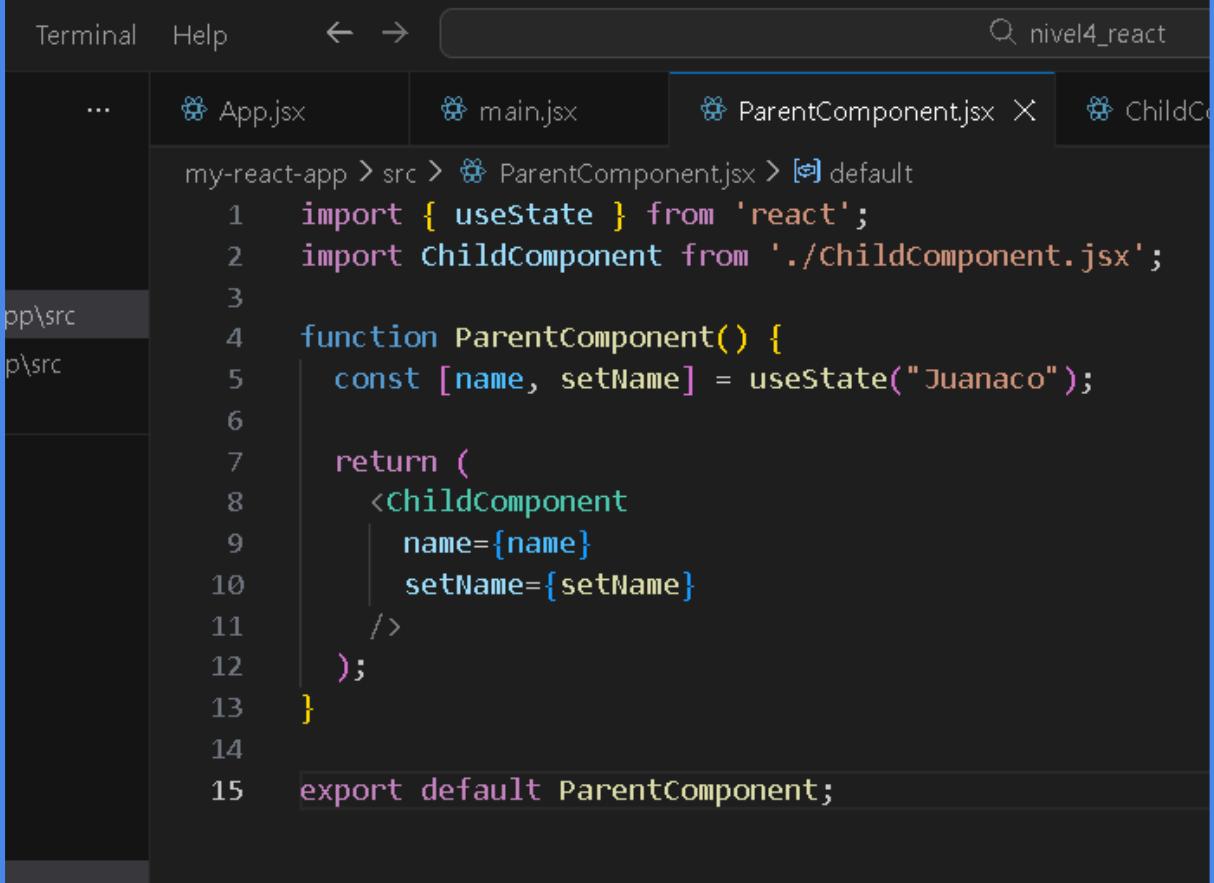
```
1 import { useState } from 'react';
2
3 function ParentComponent() {
4     const [name, setName] = useState("Juanaco");
5
6     return (
7         <>
8             <h1>Hola chavales {name}</h1>
9             <button onClick={() => setName("Juanky")}>
10                 Change Name
11             </button>
12         </>
13     );
14 }
15
16 export default ParentComponent;
```

“Figura 13, Componente padre, elaboración propia”



“Figura 14, muestra en el navegador, elaboración propia”

Parte F: Pasar estado y setState a un componente hijo



The screenshot shows a terminal window with the following details:

- Terminal tab is active.
- File tabs include App.jsx, main.jsx, ParentComponent.jsx (highlighted), and ChildComponent.jsx.
- Search bar: nivel4_react
- File structure sidebar: pp\src, p\src
- Code content:

```
my-react-app > src > ParentComponent.jsx > default
1 import { useState } from 'react';
2 import ChildComponent from './ChildComponent.jsx';
3
4 function ParentComponent() {
5   const [name, setName] = useState("Juanaco");
6
7   return (
8     <ChildComponent
9       name={name}
10      setName={setName}
11    />
12  );
13}
14
15 export default ParentComponent;
```

“Figura 15, Componente padre, elaboración propia”

The screenshot shows a code editor window with the title bar "nivel4_react". Below the title bar, there are tabs for "App.jsx", "main.jsx", "ParentComponent.jsx", and "ChildComponent.jsx", where "ChildComponent.jsx" is the active tab. The code in the editor is as follows:

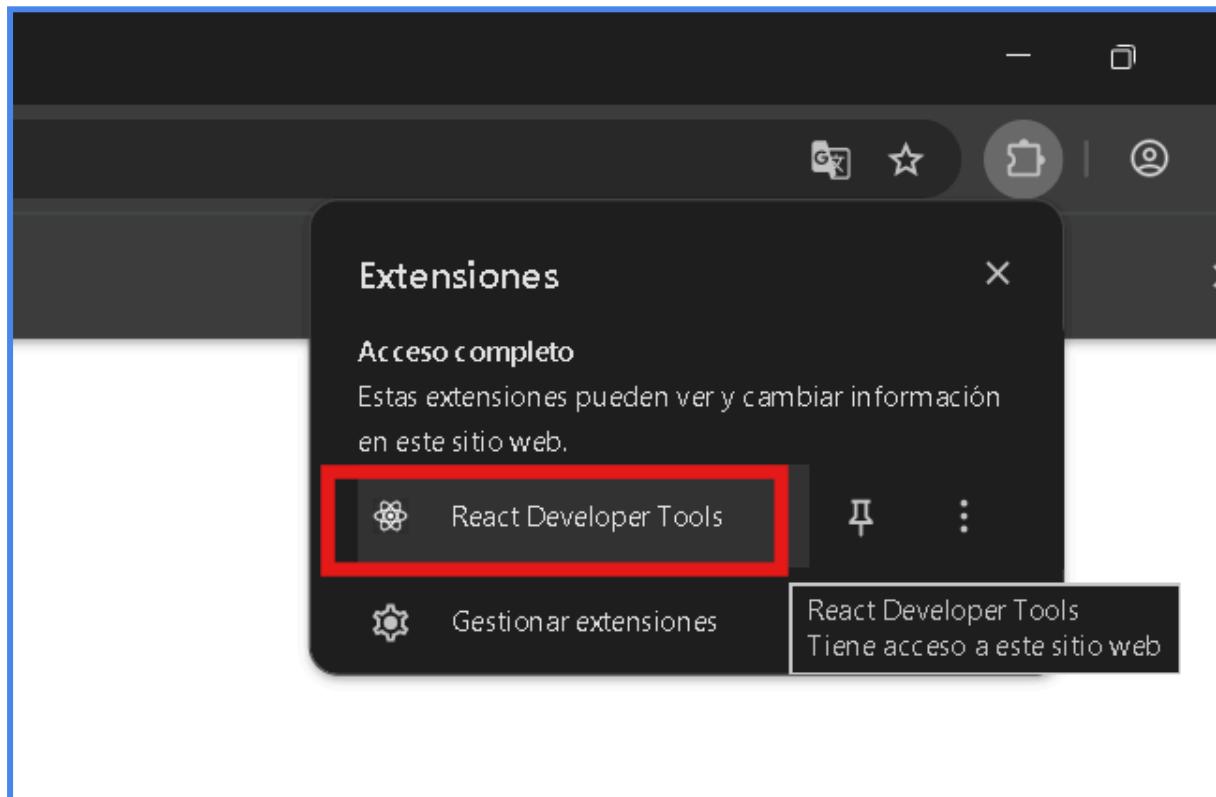
```
my-react-app > src > ChildComponent.jsx > default
1  function ChildComponent(props) {
2    return (
3      <div>
4        <h1>Hello {props.name}</h1>
5
6        <button onClick={() => props.setName("Juanky")}>
7          Poner nombre Juanky
8        </button>
9
10       <button onClick={() => props.setName("El Bicho")}>
11         Poner nombre El Bicho
12       </button>
13     </div>
14   );
15 }
16
17 export default ChildComponent;
```

“Figura 16, Componente hijo, elaboración propia”

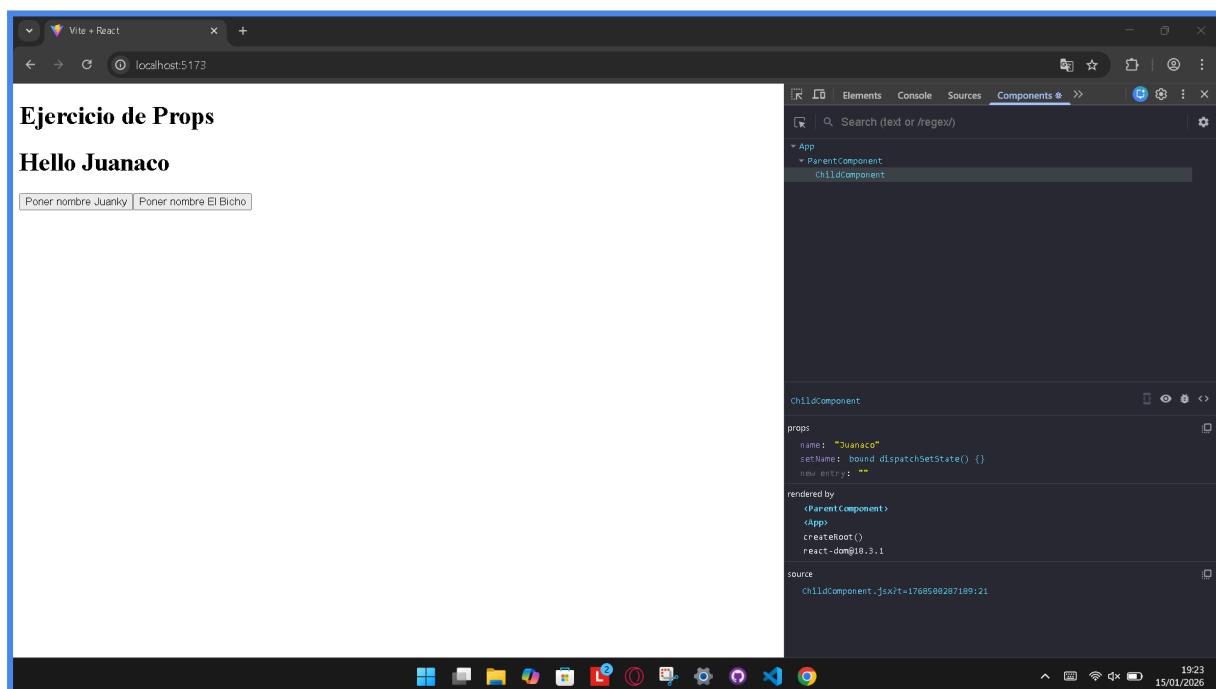


“Figura 17, muestra en el navegador, elaboración propia”

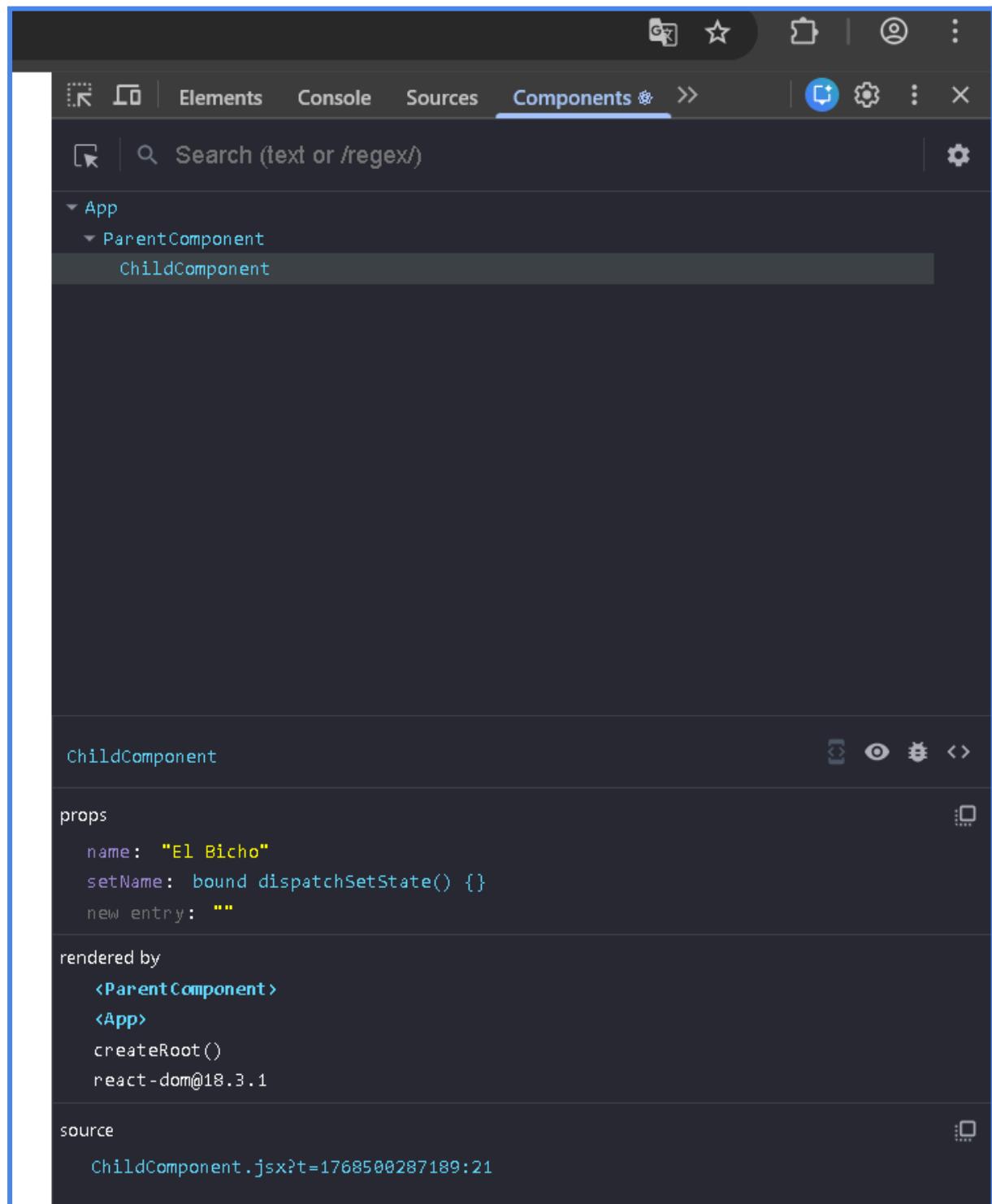
Parte G: Inspeccionar props y state con React DevTools



“Figura 18, Instalación de react developer tool, elaboración propia”



“Figura 19, muestra en el navegador con inspeccionar, elaboración propia”



“Figura 20, cambio de nombre con state actualizado, elaboración propia”