

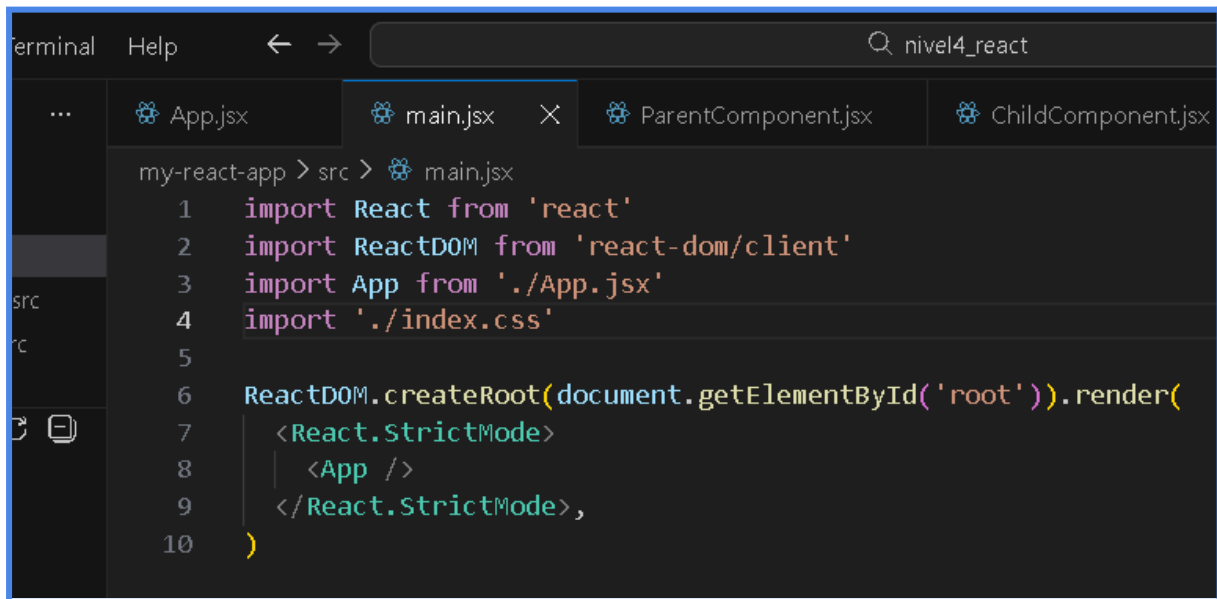
Nivel 4 React

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

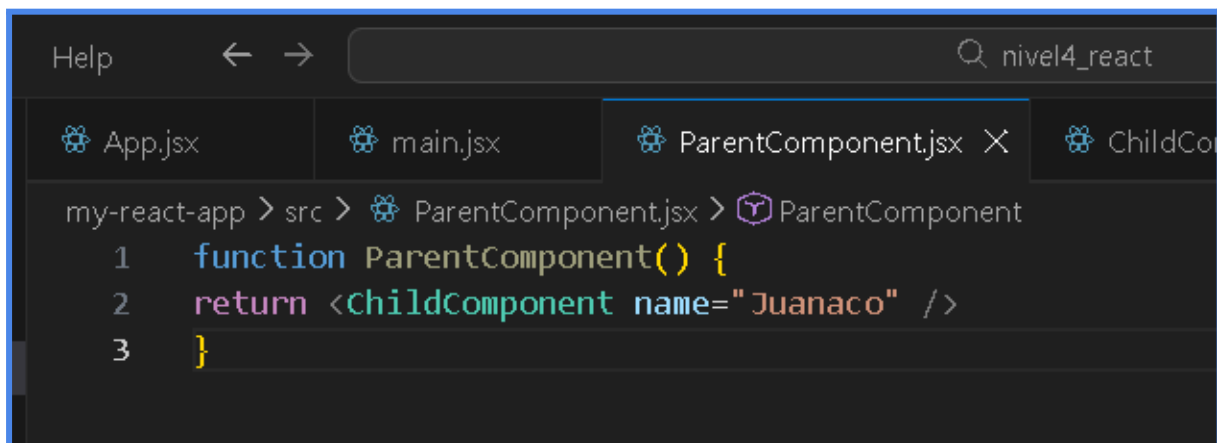
Parte A: Props básicas (Padre -> Hijo)



The screenshot shows the VS Code editor with the file explorer on the left and the editor window displaying `main.jsx`. The breadcrumb navigation shows the path: `my-react-app > src > main.jsx`. The code in `main.jsx` is as follows:

```
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”



The screenshot shows the VS Code editor with the file explorer on the left and the editor window displaying `ParentComponent.jsx`. The breadcrumb navigation shows the path: `my-react-app > src > ParentComponent.jsx > ParentComponent`. The code in `ParentComponent.jsx` is as follows:

```
1 function ParentComponent() {
2   return <ChildComponent name="Juanaco" />
3 }
```

“Figura 02, componente padre, elaboración propia”

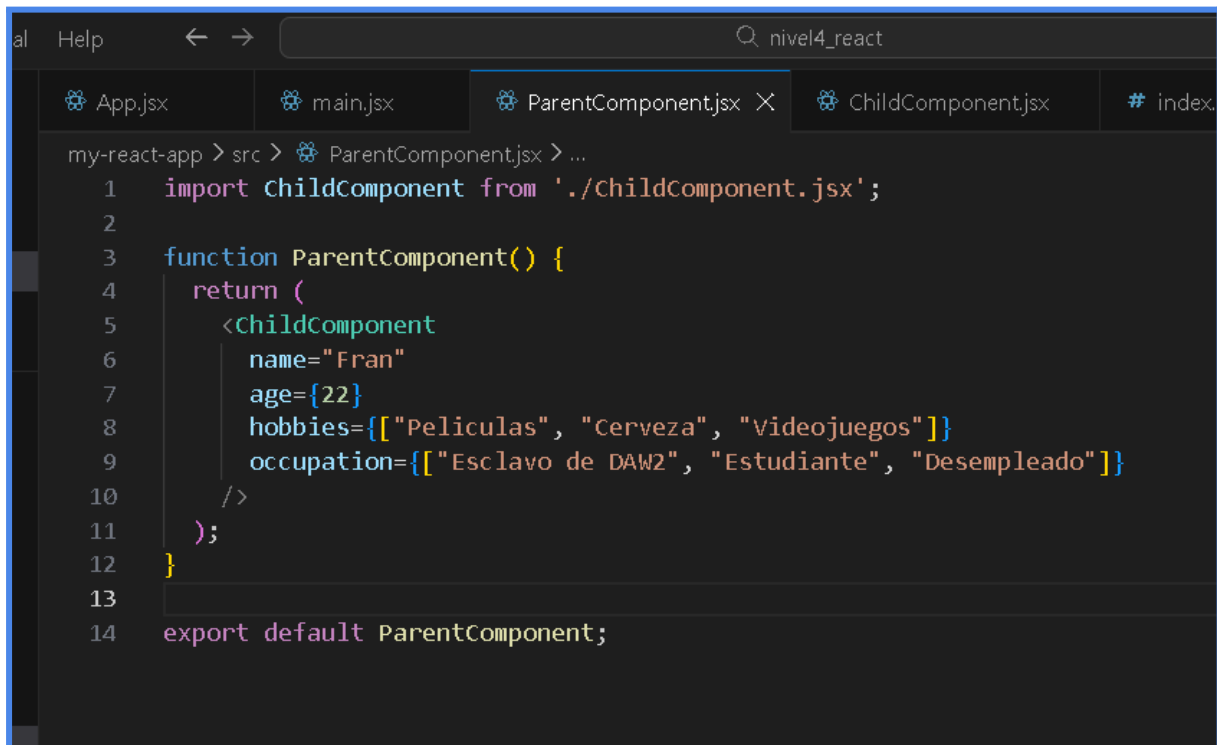


The screenshot shows the VS Code editor with the file explorer on the left and the editor window displaying `ChildComponent.jsx`. The breadcrumb navigation shows the path: `my-react-app > src > ChildComponent.jsx > ChildComponent`. The code in `ChildComponent.jsx` is as follows:

```
1 function ChildComponent(props) {
2   return <p>Hello bros! my name is {props.name}</p>
3 }
```

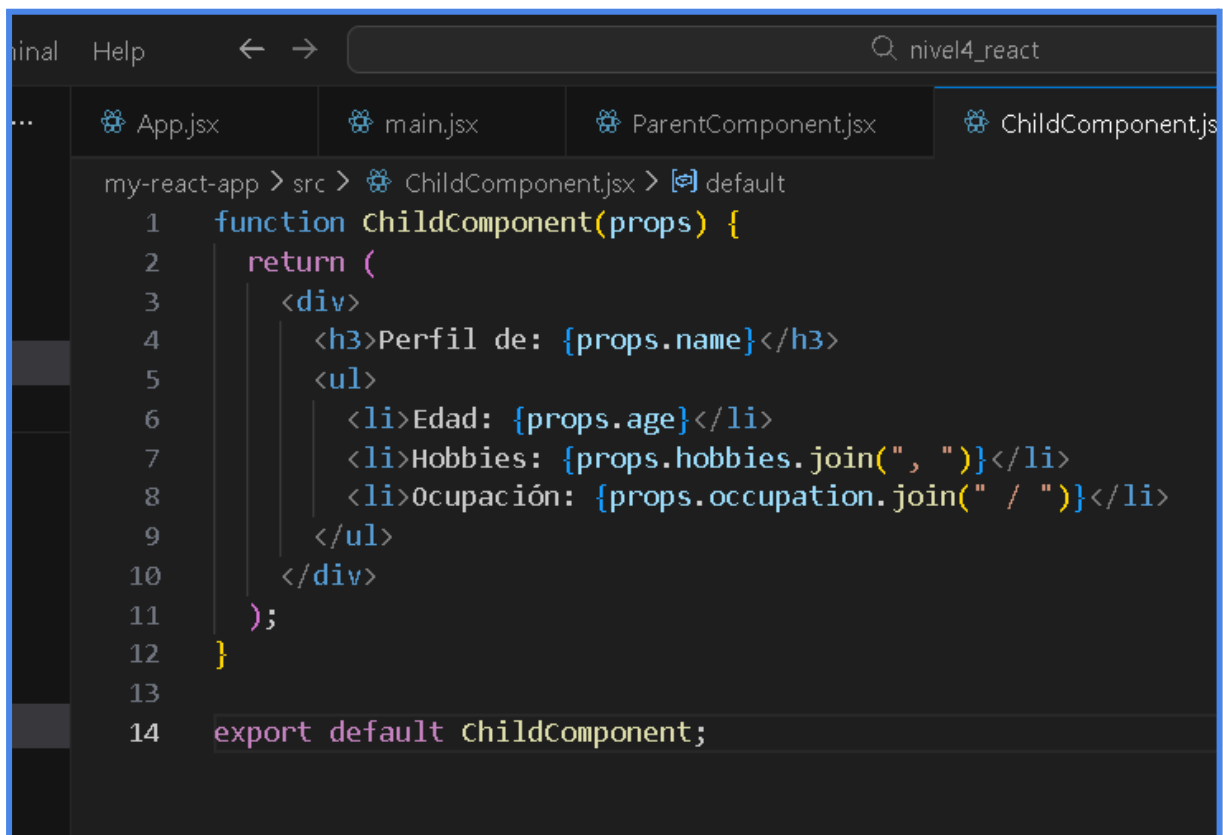
“Figura 03, componente hijo, elaboración propia”

Parte B: Pasar múltiples props



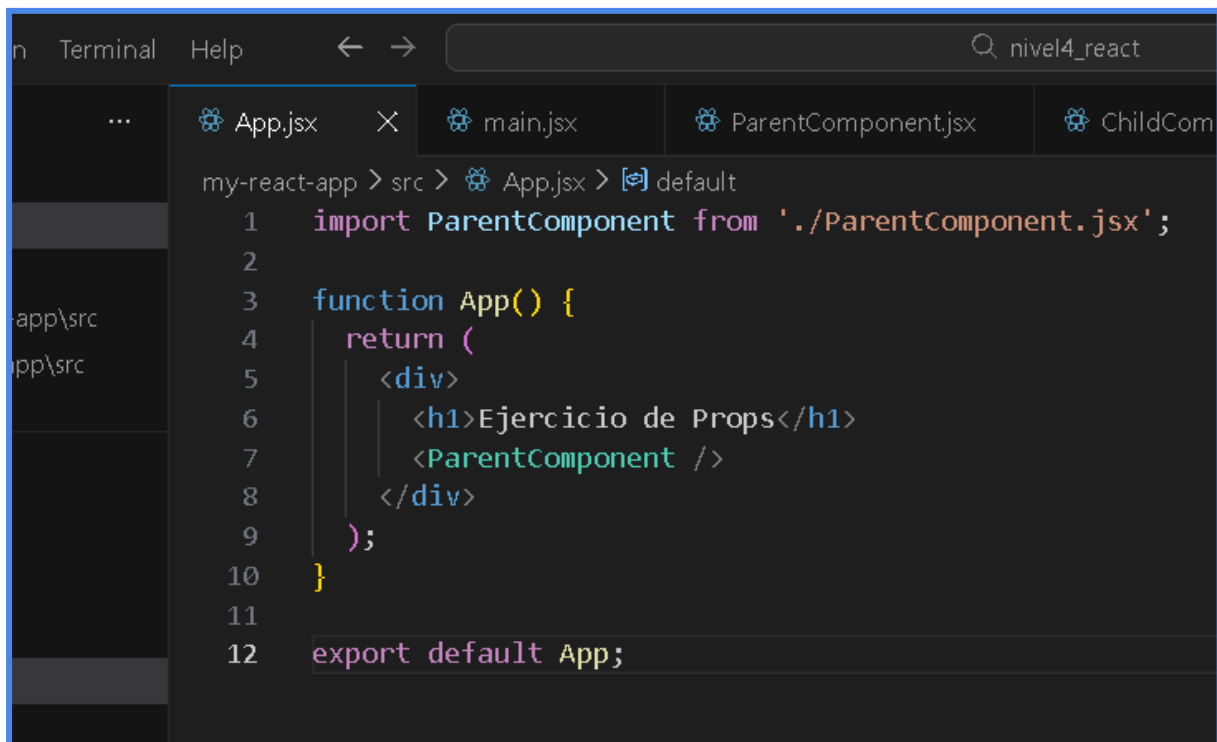
```
al Help ← → nivel4_react
App.jsx main.jsx ParentComponent.jsx X ChildComponent.jsx # index
my-react-app > src > ParentComponent.jsx > ...
1 import ChildComponent from './ChildComponent.jsx';
2
3 function ParentComponent() {
4   return (
5     <ChildComponent
6       name="Fran"
7       age={22}
8       hobbies=["Películas", "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”



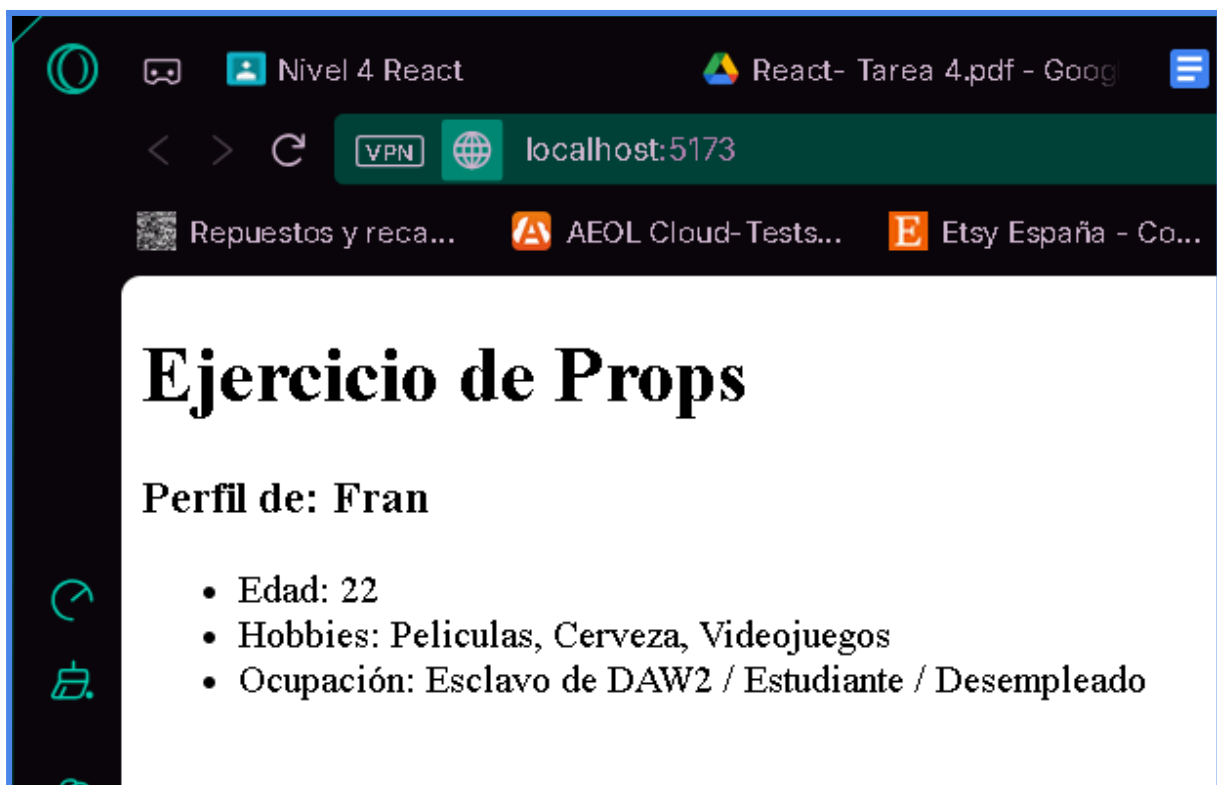
```
inal Help ← → nivel4_react
... App.jsx main.jsx ParentComponent.jsx ChildComponent.jsx
my-react-app > src > ChildComponent.jsx > default
1 function ChildComponent(props) {
2   return (
3     <div>
4       <h3>Perfil de: {props.name}</h3>
5       <ul>
6         <li>Edad: {props.age}</li>
7         <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”



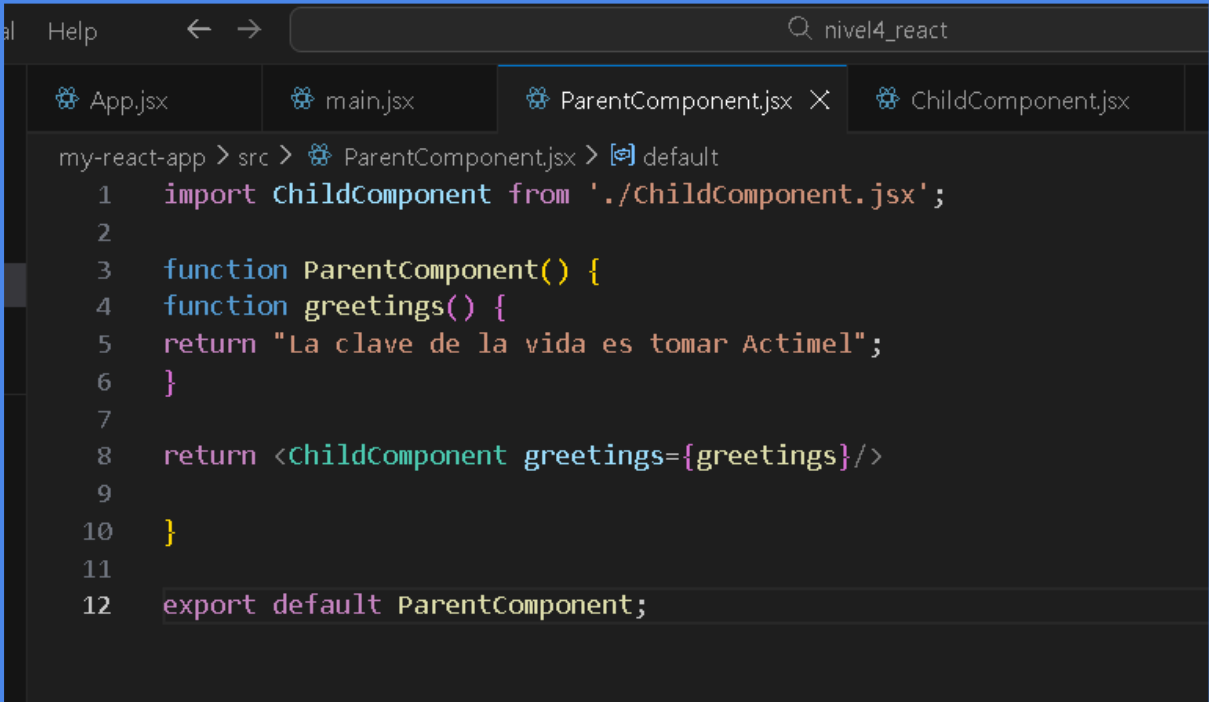
```
my-react-app > src > App.jsx > default
1  import ParentComponent from './ParentComponent.jsx';
2
3  function App() {
4    return (
5      <div>
6        <h1>Ejercicio de Props</h1>
7        <ParentComponent />
8      </div>
9    );
10 }
11
12 export default App;
```

“Figura 06, app.jsx, elaboración propia”



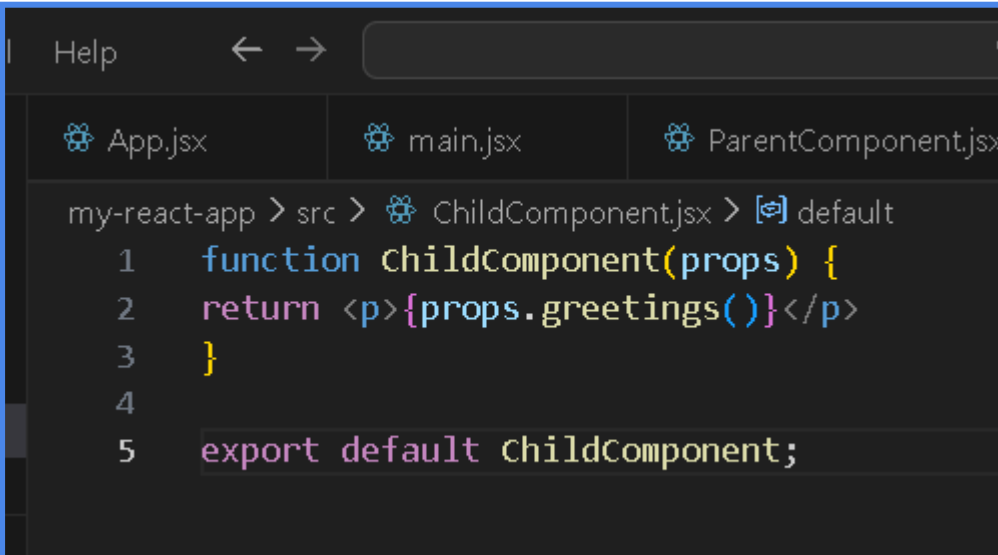
“Figura 07, muestra en el navegador, elaboración propia”

Parte C: Pasar funciones como props



```
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”



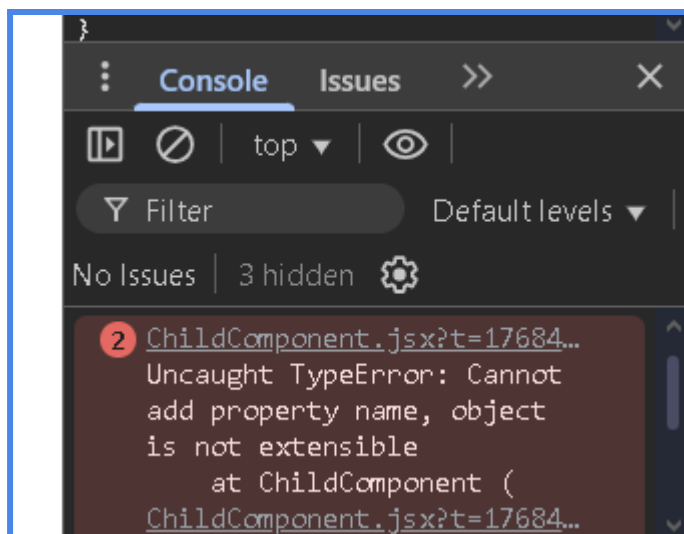
```
my-react-app > src > ChildComponent.jsx > default
1  function ChildComponent(props) {
2    return <p>{props.greetings()}</p>
3  }
4
5  export default childComponent;
```

“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 asique he tenido que acceder a la consola del inspeccionar, elaboración propia”

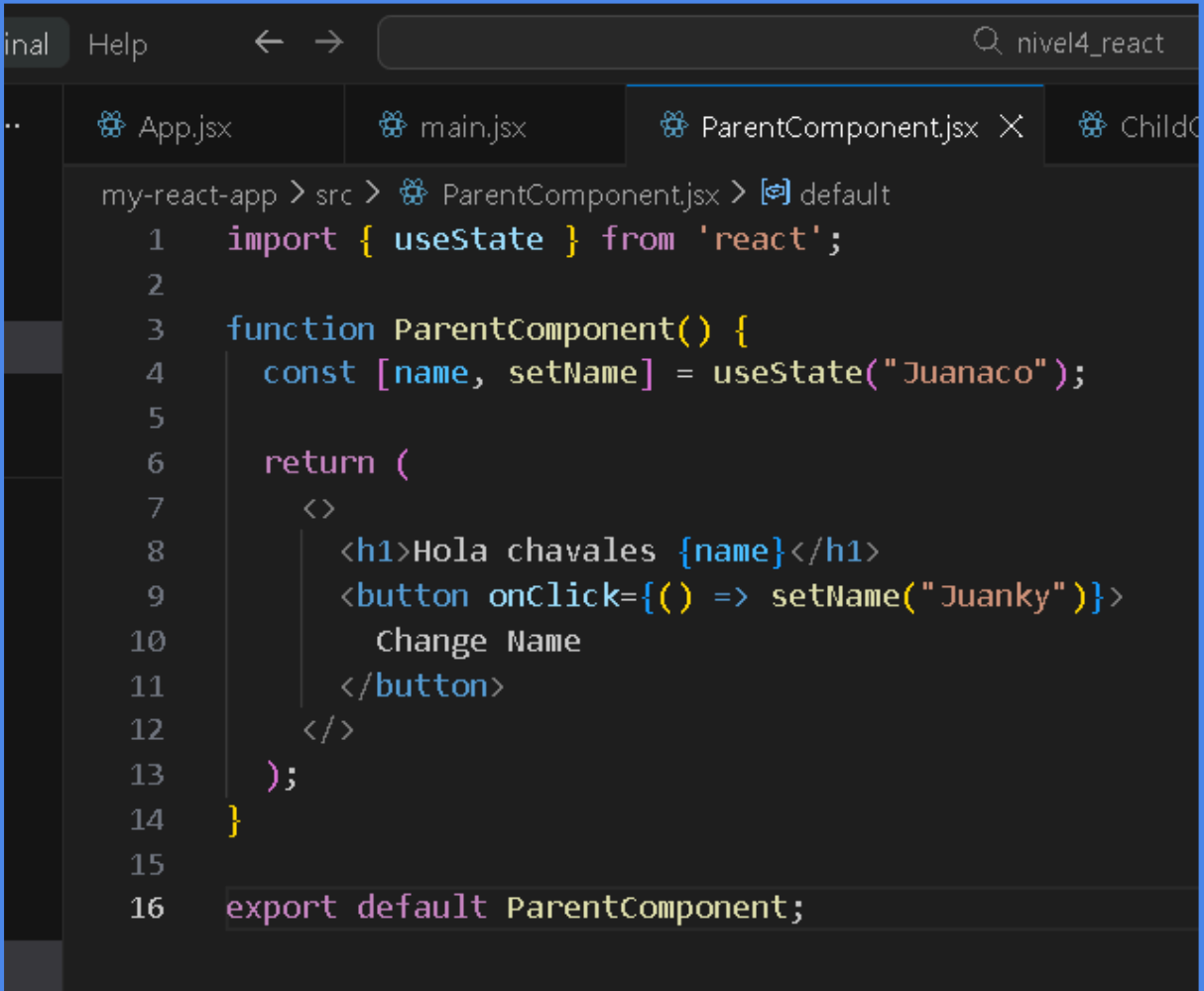


The image shows a code editor window with a dark theme. At the top, there's a search bar with the text "nivel4_react". Below it, a tab bar shows four files: "App.jsx", "main.jsx", "ParentComponent.jsx", and "ChildComponent.jsx". The "ChildComponent.jsx" tab is selected. The editor displays the following 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 image shows a code editor window with a dark theme. The top bar includes a search icon and the text "nivel4_react". Below the top bar, there are tabs for "App.jsx", "main.jsx", "ParentComponent.jsx", and "ChildC...". The "ParentComponent.jsx" tab is active. The code in the editor is as follows:

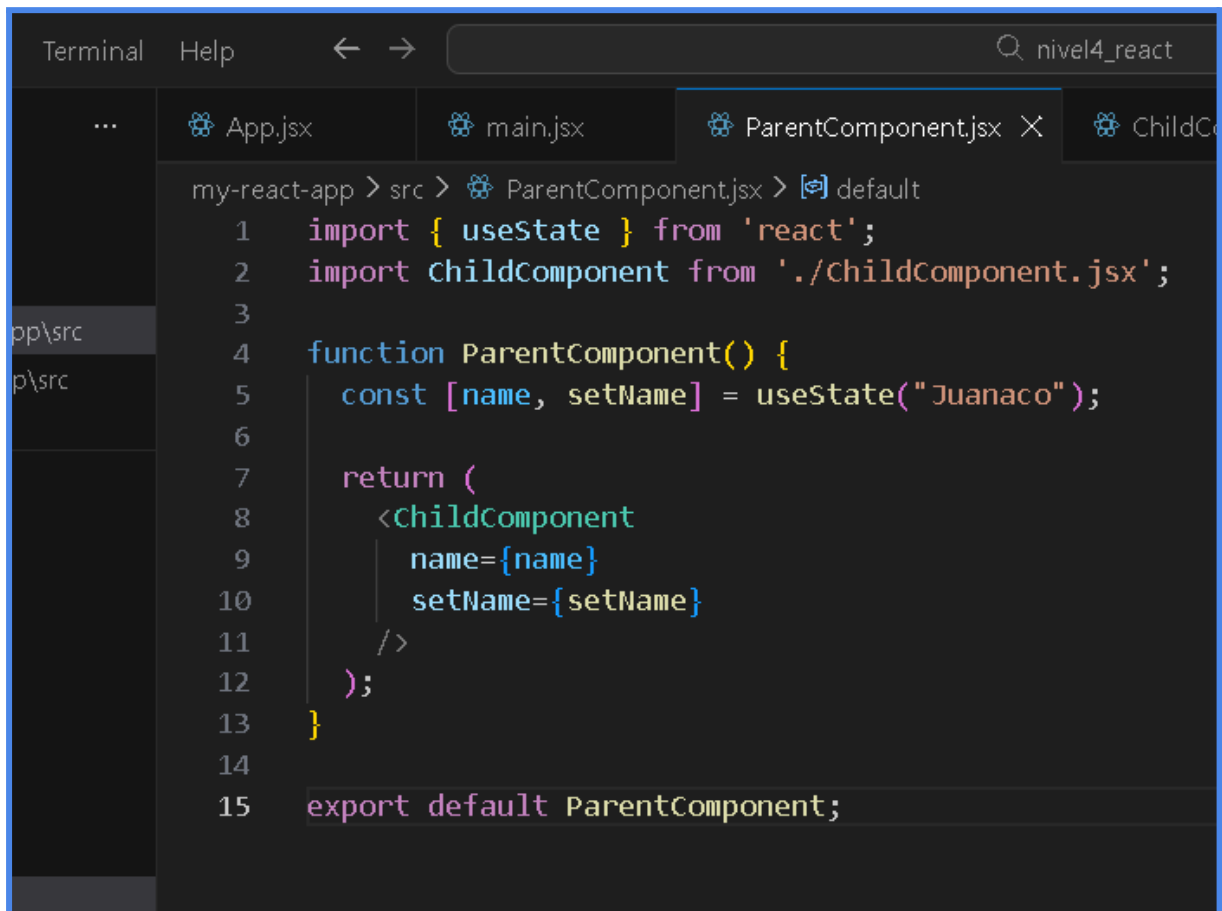
```
my-react-app > src > ParentComponent.jsx > [icon] default
1  import { useState } from 'react';
2
3  function ParentComponent() {
4    const [name, setName] = useState("Juanaco");
5
6    return (
7      <>
8        <h1>Hola chavales {name}</h1>
9        <button onClick={() => setName("Juanky")}>
10          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

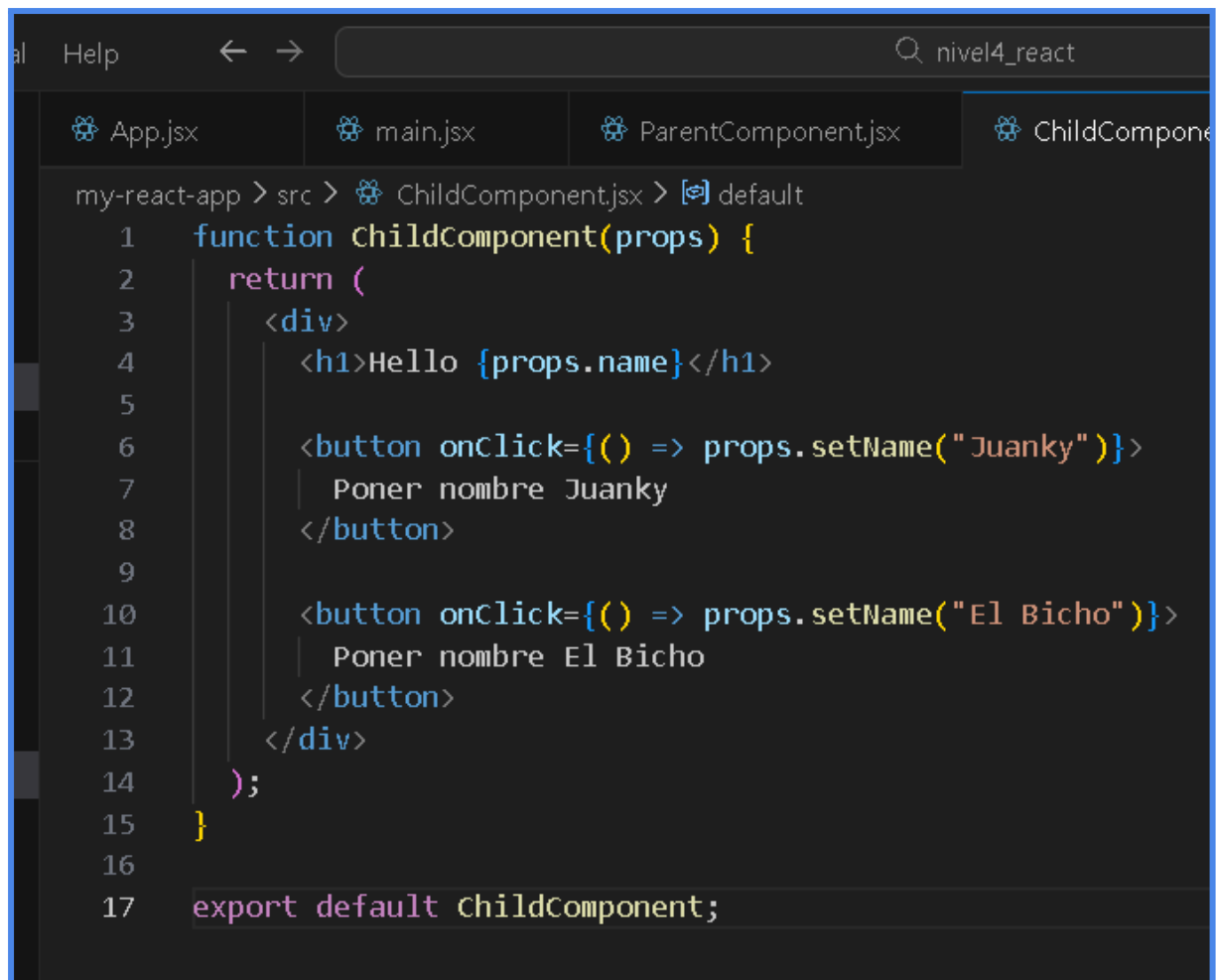


```
Terminal  Help  <  >  nivel4_react

App.jsx  main.jsx  ParentComponent.jsx  ChildComponent.jsx

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 with a dark theme. The breadcrumb path at the top reads: `my-react-app > src > ChildComponent.jsx > default`. The editor displays the following code:

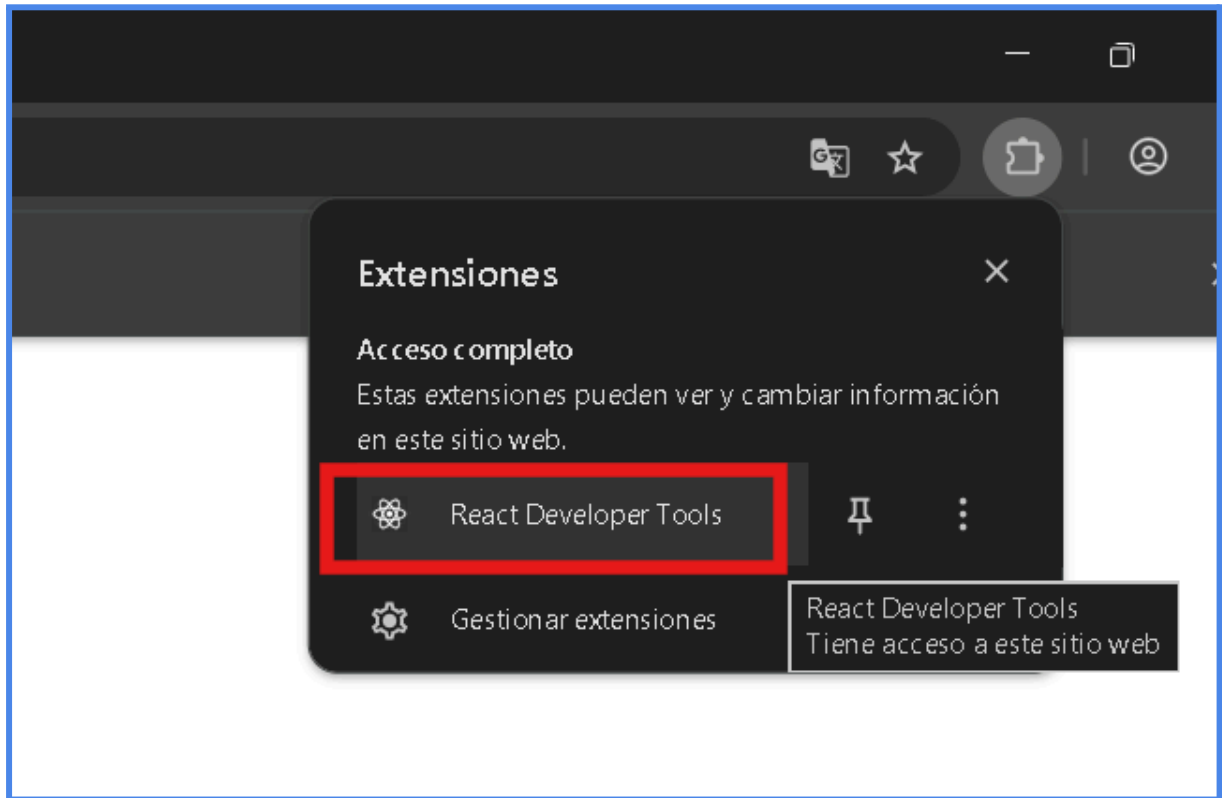
```
1 function childComponent(props) {
2   return (
3     <div>
4       <h1>Hello {props.name}</h1>
5
6       <button onClick={() => props.setName("Juanky")}>
7         Poner nombre Juanky
8       </button>
9
10      <button onClick={() => props.setName("El Bicho")}>
11        Poner nombre El Bicho
12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

“Figura 16, Componente hijo, elaboración propia”

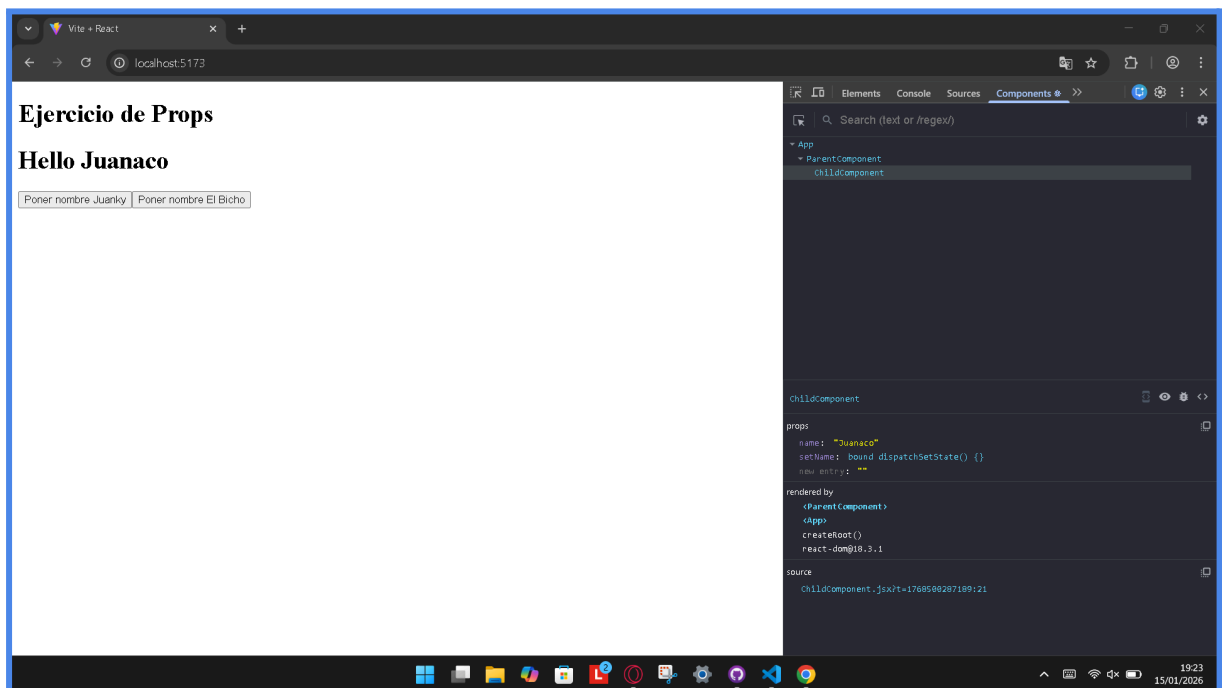


“Figura 17, muestra en el navegador, elaboración propia”

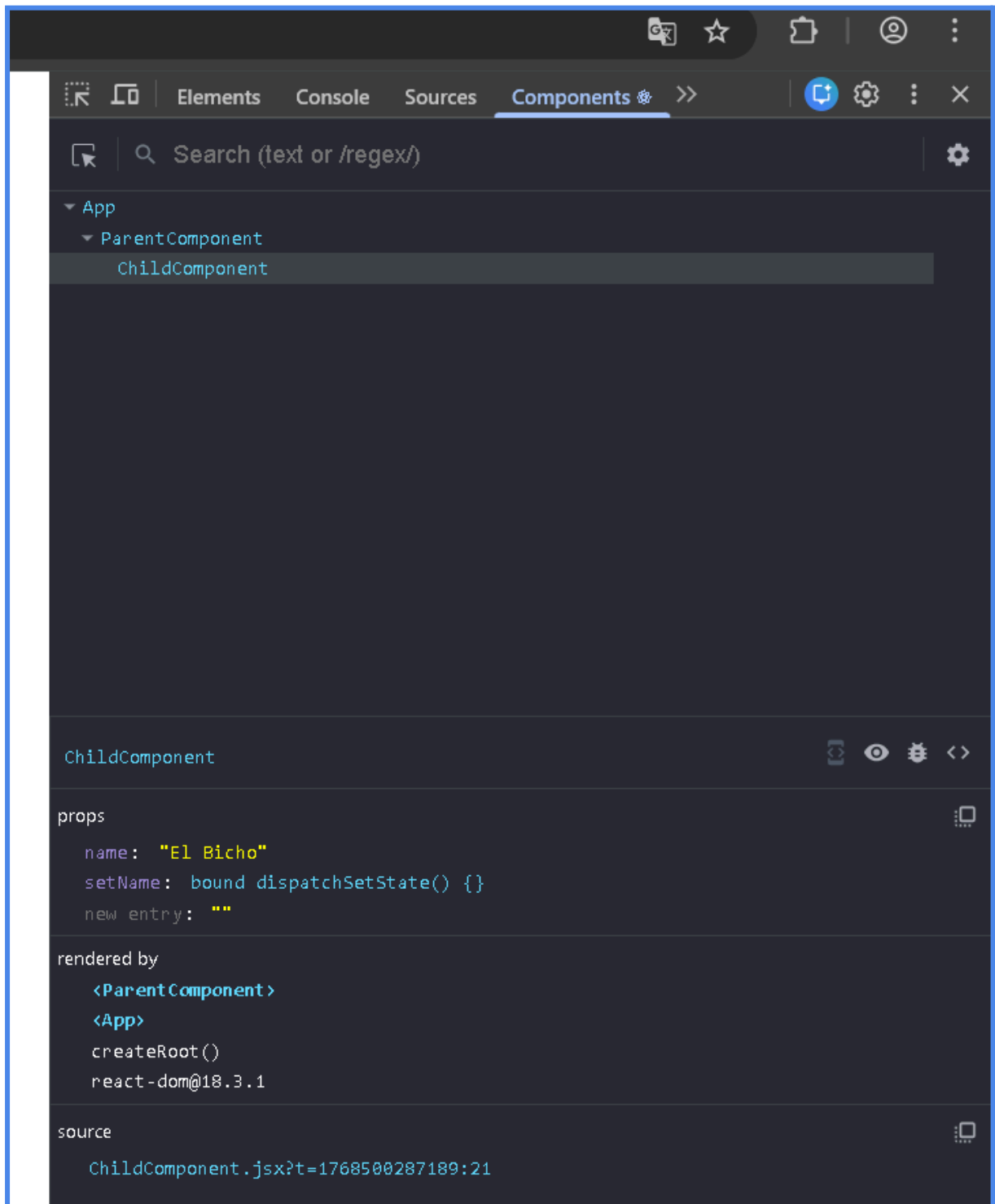
Parte G: Inspeccionar props y state con React DevTools



“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”