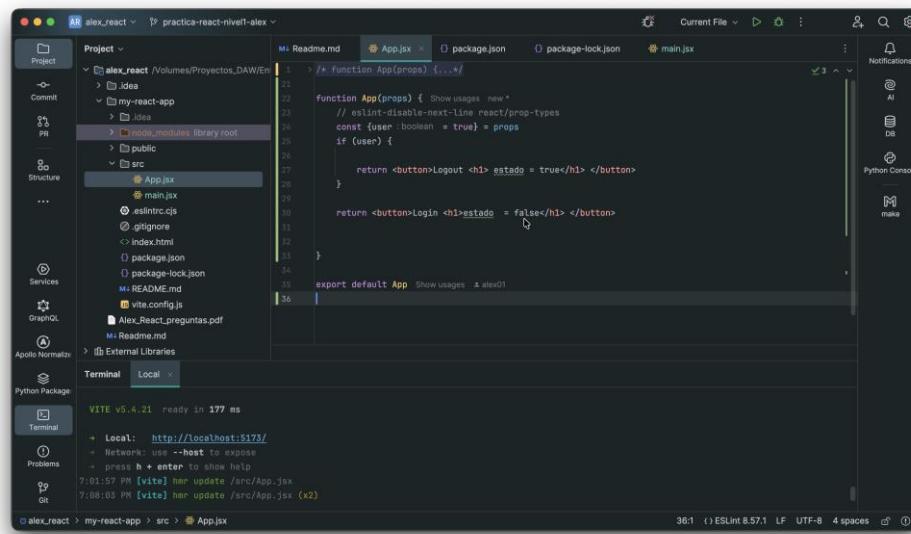


1- Parte A: Condicional con if + return

He creado un h1, para saber en qué estado esta del usuario
 Lo he creado simplemente para que sea visual

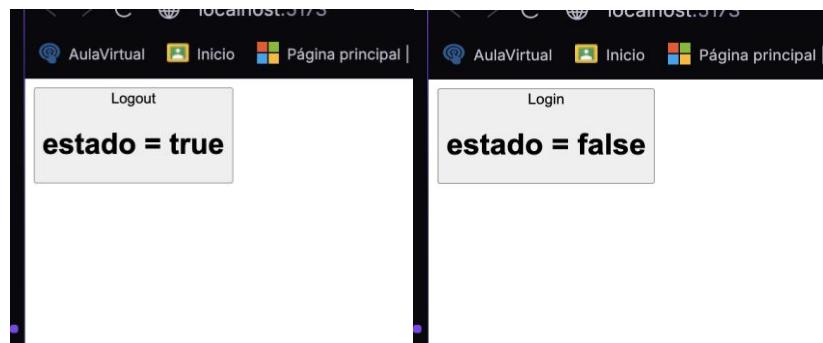


```

function App(props) {
  const user = true;
  if (user) {
    return <button>Logout <h1>estado = true</h1> </button>
  }
  return <button>Login <h1>estado = false</h1> </button>
}

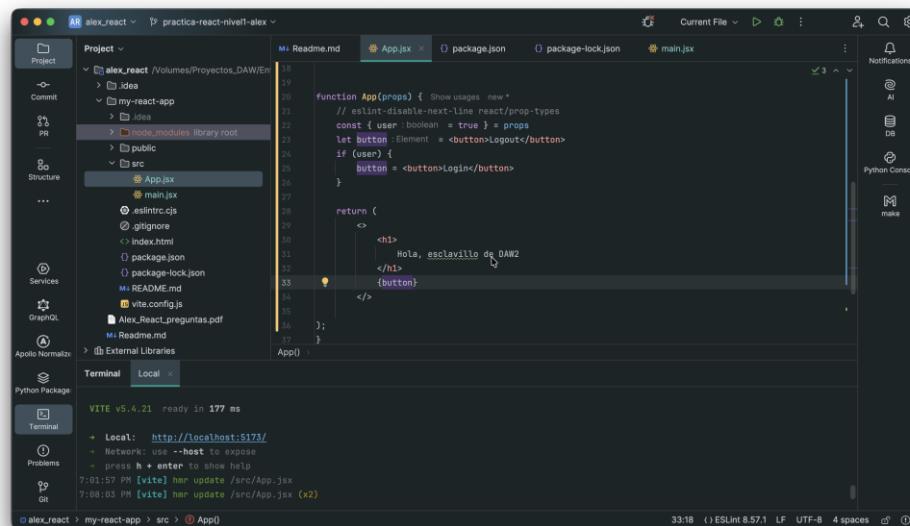
export default App
  
```

Resultado



2- Parte B: Renderizado parcial usando una variable

Se ha creado la variable button, para luego en el Return ser llamada



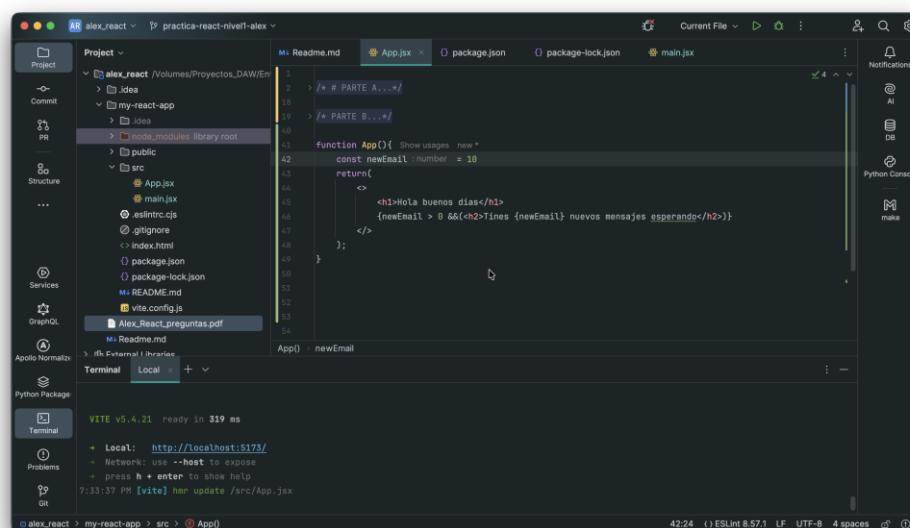
The screenshot shows the VS Code interface with the project structure on the left and the code editor on the right. The code editor displays the following App.js content:

```
function App(props) { Show usages new *  
  // eslint-disable-next-line react/prop-types  
  const [ user :boolean = true ] = props  
  let button :Element = <button>Logout</button>  
  if (user) {  
    button = <button>Login</button>  
  }  
  
  return (  
    <>  
      <h1>  
        Hola, esclavillo de DAW2  
      </h1>  
      {button}  
    </>  
  );  
};
```

The terminal at the bottom shows the command `yarn update` being run.

3- Parte C: Renderizado en línea con el operador &&

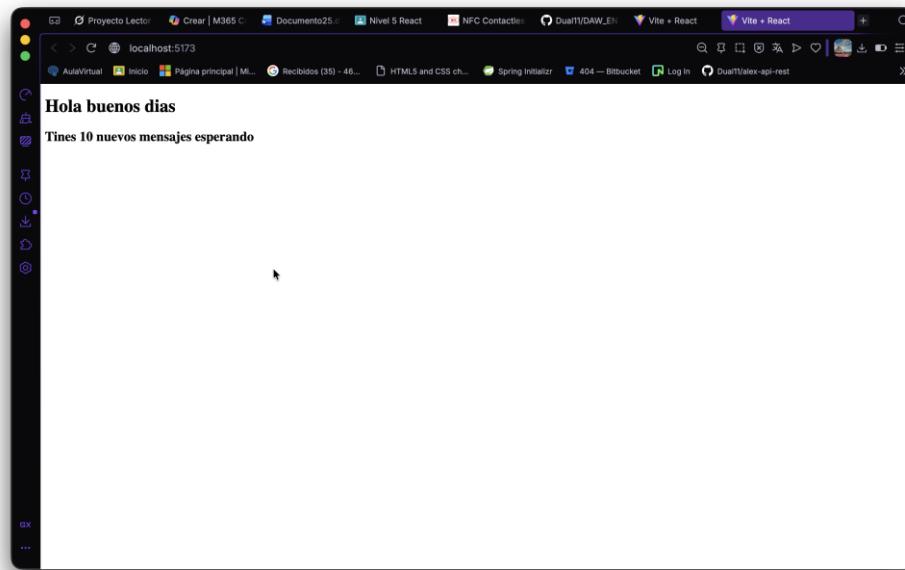
He modificado la cantidad en la variable para luego mostrarlo por pantalla



The screenshot shows the VS Code interface with the project structure on the left and the code editor on the right. The code editor displays the following App.js content:

```
> /* # PARTE A... */  
> /* PARTE B... */  
  
function App(){ Show usages new *  
  const newEmail :number = 10  
  return(  
    <>  
      <h1>Hola buenos dias</h1>  
      {newEmail > 0 &&(<h2>Nuevos mensajes esperando</h2>)}  
    </>  
  );  
};
```

The terminal at the bottom shows the command `yarn update` being run.



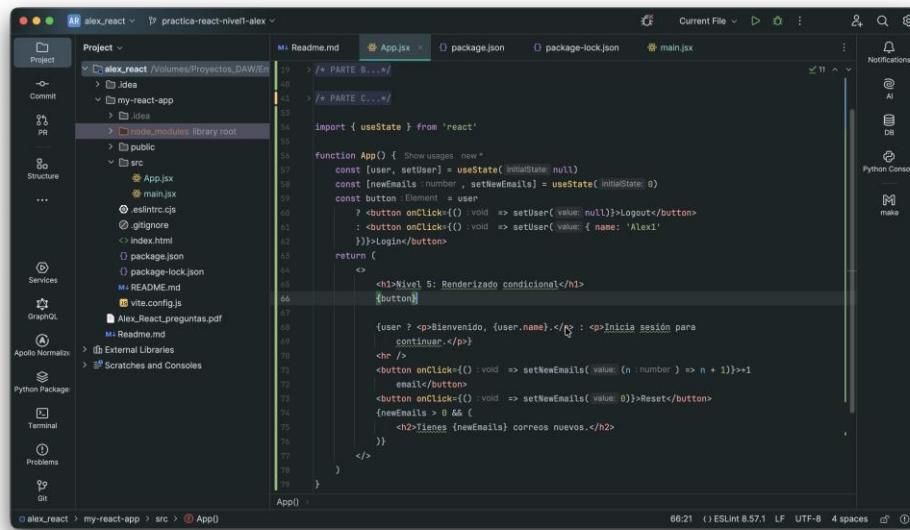
4- Parte D: Renderizado en línea con operador ternario

El mensaje cambia, en base al estado del Usuario si es false o true

```
1 > /* # PARTE A...*/
2 // eslint-disable-next-line react/prop-types
3 const { user :boolean = false } = props;
4
5 return (
6   <>
7     <h1>Hola chavul@s</h1>
8     {user ? <button>Logout</button> : <button>Login</button>}
9   </>
10 )
11 }
12
13 export default App Show usages ▾ alex01
14
```

VITE v5.4.21 ready in 288 ms
Local: http://localhost:5173/
Network: use --host to expose
press h + enter to show help

5- Mini reto

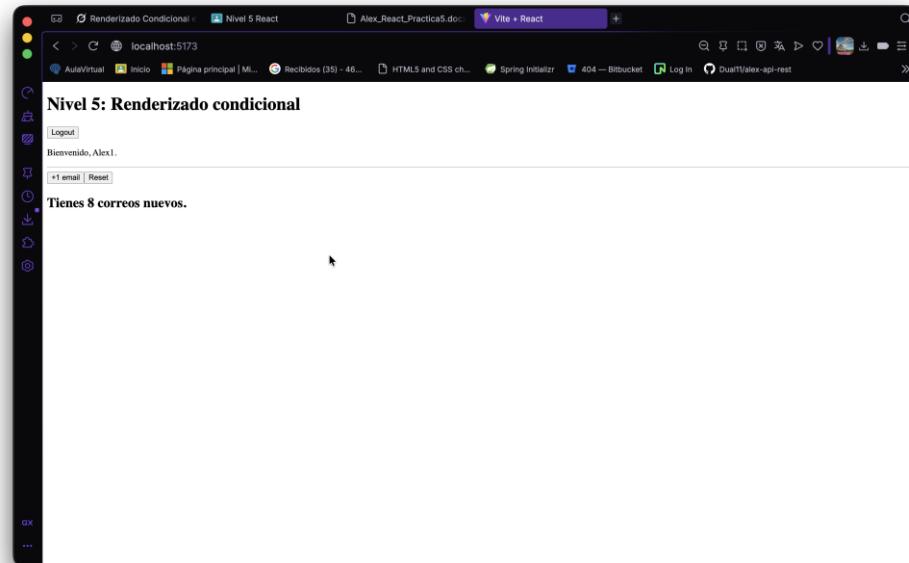


The screenshot shows the VS Code interface with the file `App.jsx` open. The code implements a conditional rendering logic based on the user's presence:

```
import { useState } from 'react';

function App() {
  const [user, setUser] = useState(null);
  const [newEmails, setNewEmails] = useState(0);
  const button = () => user ? (
    <button onClick={()=> setUser(null)}>Logout</button>
    : <button onClick={()=> setUser({ name: 'Alex' })}>Login</button>
  );
  return (
    <h1>Nivel 5: Renderizado condicional</h1>
    <button>
      {user ? <p>Bienvenido, {user.name}.</p> : <p>Inicia sesión para continuar.</p>}
      <br />
      <button onClick={()=> setNewEmails((n: number) => n + 1)}>+1 email</button>
      <button onClick={()=> setNewEmails(0)}>Reset</button>
    <h2>Tienes {newEmails} correos nuevos.</h2>
  );
}

export default App;
```



PREGUNTAS EXTRA

¿Cuál es la diferencia entre usar if/return y usar un ternario en JSX?

La diferencia es que if/return, es muy útil para estructuras muy diferente y elige cual elemento retornar. Mientras que el ternario permite elegir entre elementos estáticos sin múltiple return.

¿Cuándo te conviene usar && en lugar de un ternario?

Cuando se quiere mostrar u ocultar algo cuando la elección es true o false.

¿Qué ventaja tiene renderizar una parte dinámica en una variable (por ejemplo, button)?

Para que el código sea más legible y permite entenderlo mejor ya que separa la lógica, esto nos permite reutilizar el código.

Describe un caso real en una app donde el renderizado condicional sea imprescindible.

Un ejemplo simple seria cuando la acción del usuario quiera añadir o eliminar algún producto para comprar, esto permite un cambio de estado y no seria estático.