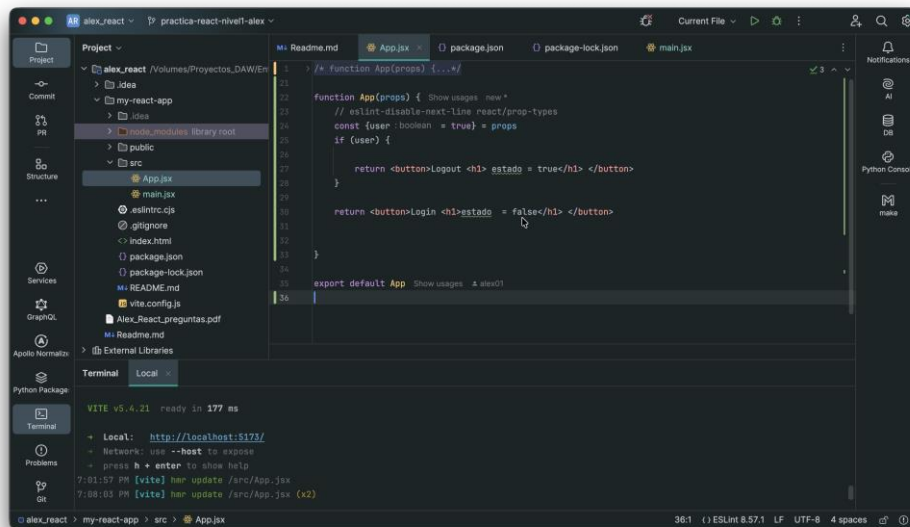


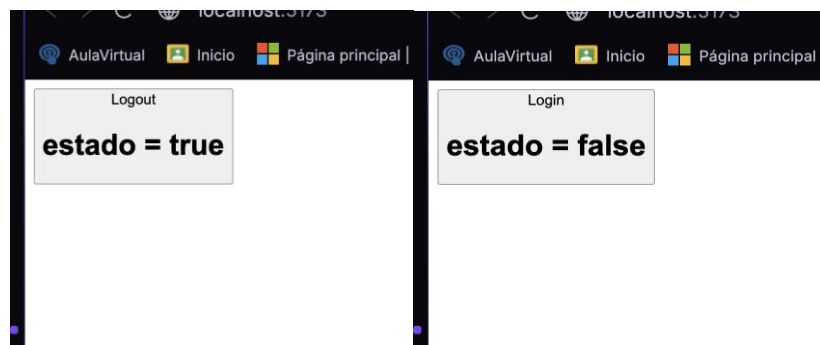
## 1- Parte A: Condicional con if + return

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



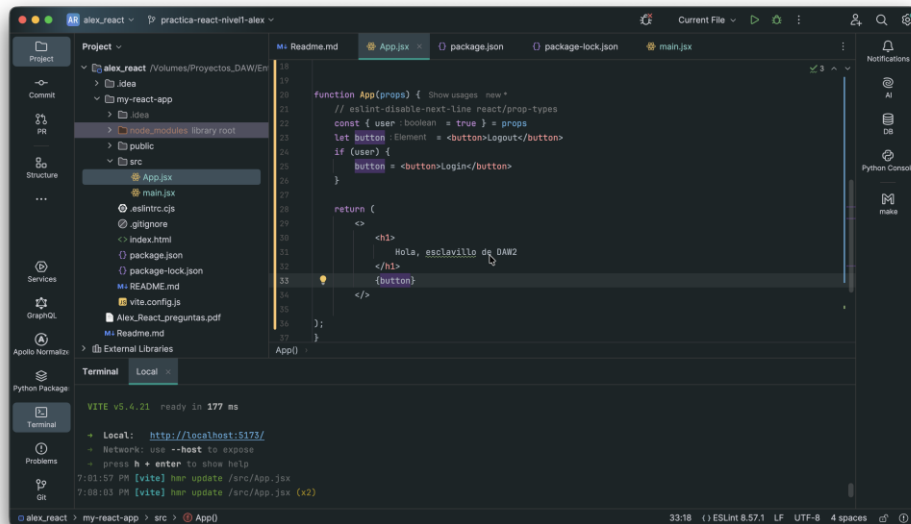
```
function App(props) {  
  // eslint-disable-next-line react/prop-types  
  const {user: boolean = true} = props  
  if (user) {  
    return <button>Logout <h1> estado = true</h1> </button>  
  }  
  return <button>Login <h1>estado = false</h1> </button>  
}  
  
export default App
```

Resultado



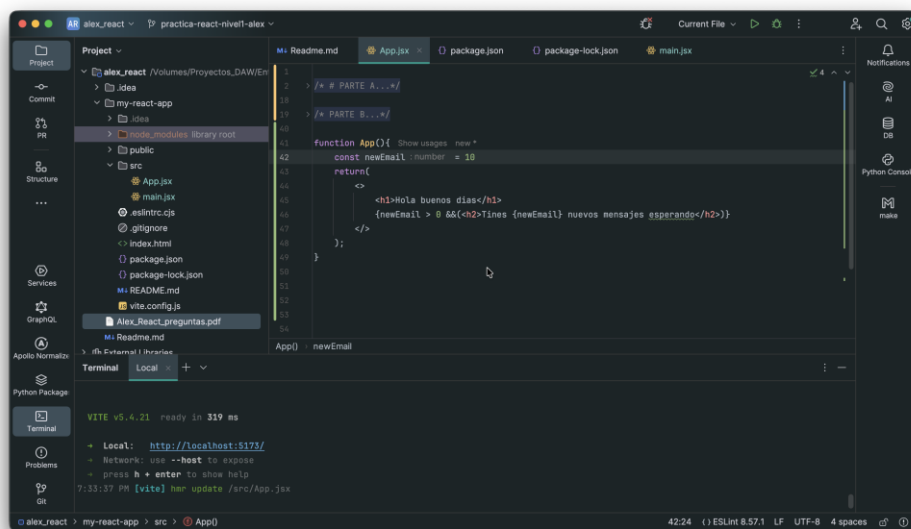
## 2- Parte B: Renderizado parcial usando una variable

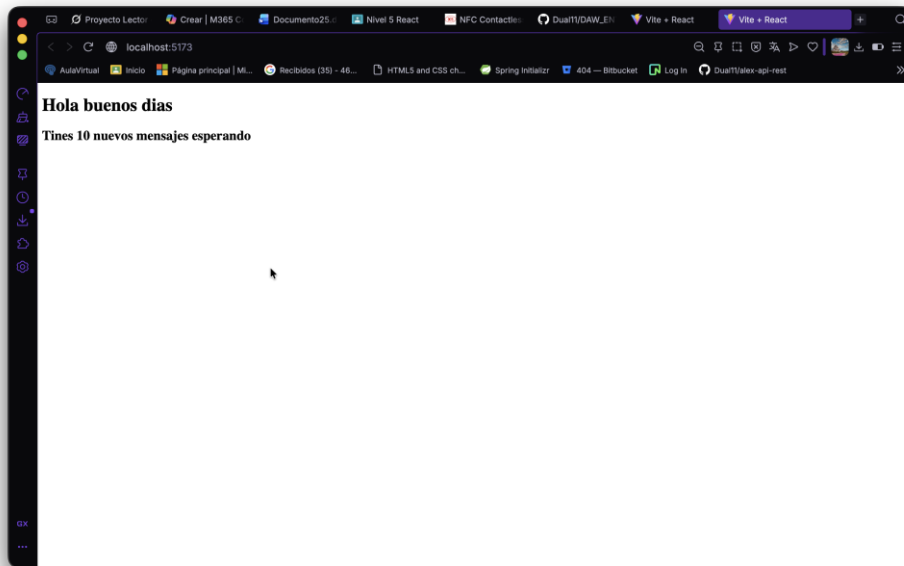
Se ha creado la variable button, para luego en el Return ser llamada



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

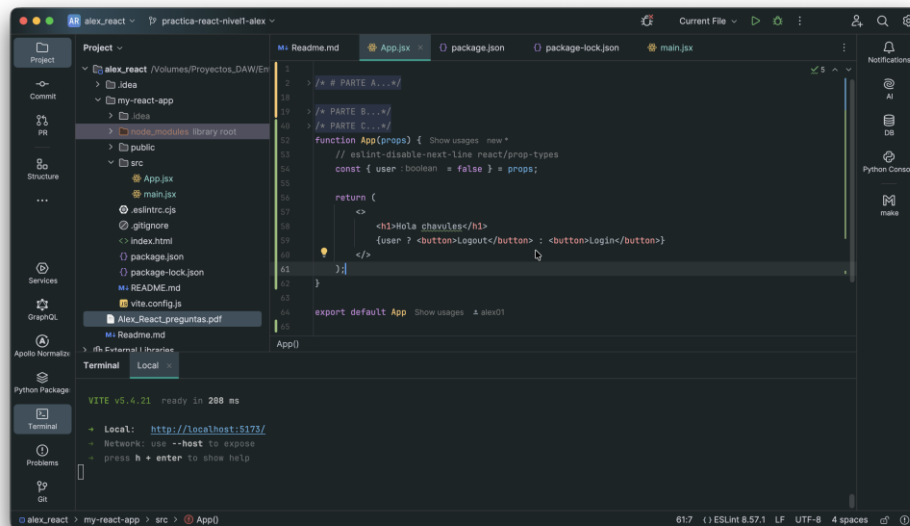
He modificado la cantidad en la variable para luego mostrarlo por pantalla



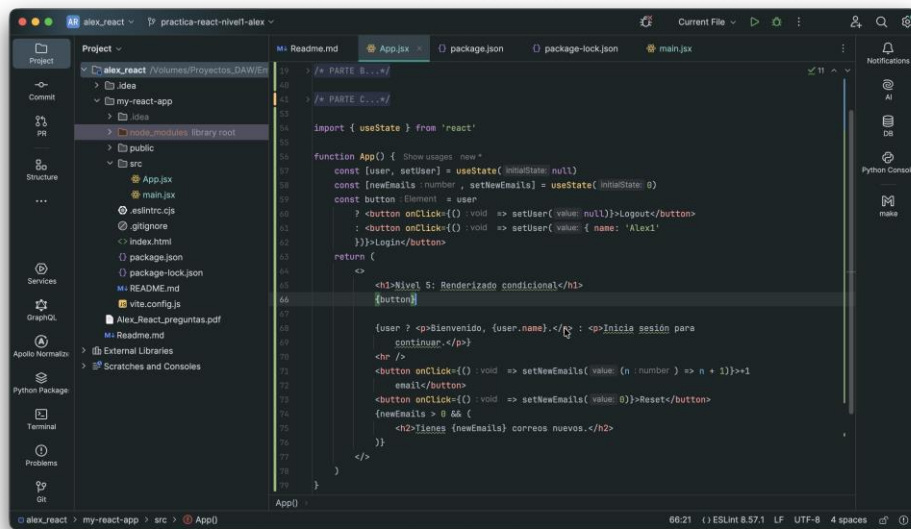


#### 4- Parte D: Renderizado en línea con operador ternario

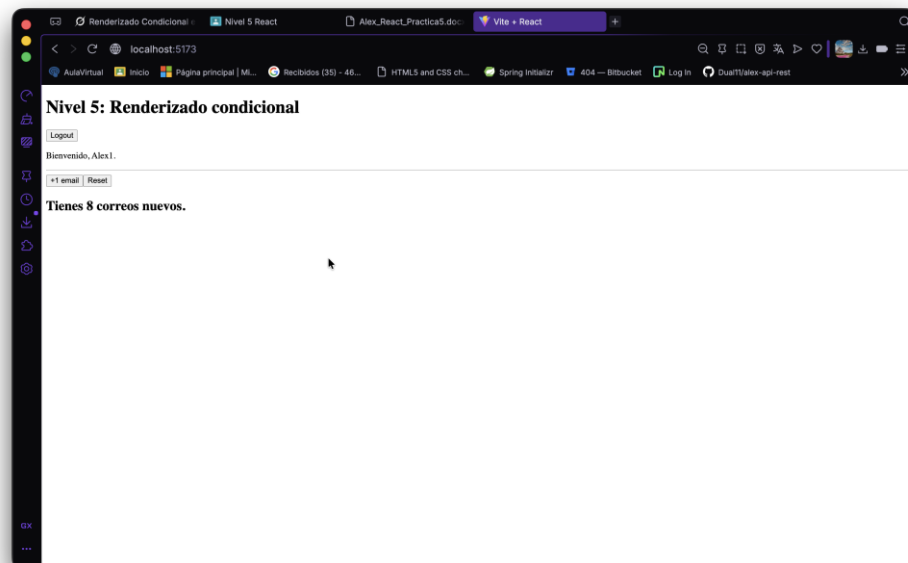
El mensaje cambia, en base al estado del Usuario si es false o true



## 5- Mini reto



```
1  // PARTE 8...
2
3  // PARTE 9...
4
5  import { useState } from 'react'
6
7  function App() {
8    const [user, setUser] = useState(initialState: null)
9    const [newEmails, setNewEmails] = useState(initialState: 0)
10    const button = <button> + user
11    ? <button onClick={() => void => setUser({value: null})}>Logout</button>
12    : <button onClick={() => void => setUser({value: 'Alex'})}>Login</button>
13    return (
14      <>
15        <h1>Nivel 5: Renderizado condicional</h1>
16        <button>
17
18        {user ? <p>Bienvenido, {user.name}</p> : <p>Inicia sesión para
19          continuar.</p>}
20        </>
21        <button onClick={() => void => setNewEmails({value: (n: number) => n + 1})>+1
22          email</button>
23        <button onClick={() => void => setNewEmails({value: 0})>Reset</button>
24        {newEmails > 0 && (
25          <h2>Tienes {newEmails} correos nuevos.</h2>
26        )}
27      </>
28    )
29  }
30
31  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 diferentes 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 sería cuando la acción del usuario quiera añadir o eliminar algún producto para comprar, esto permite un cambio de estado y no sería estático.