

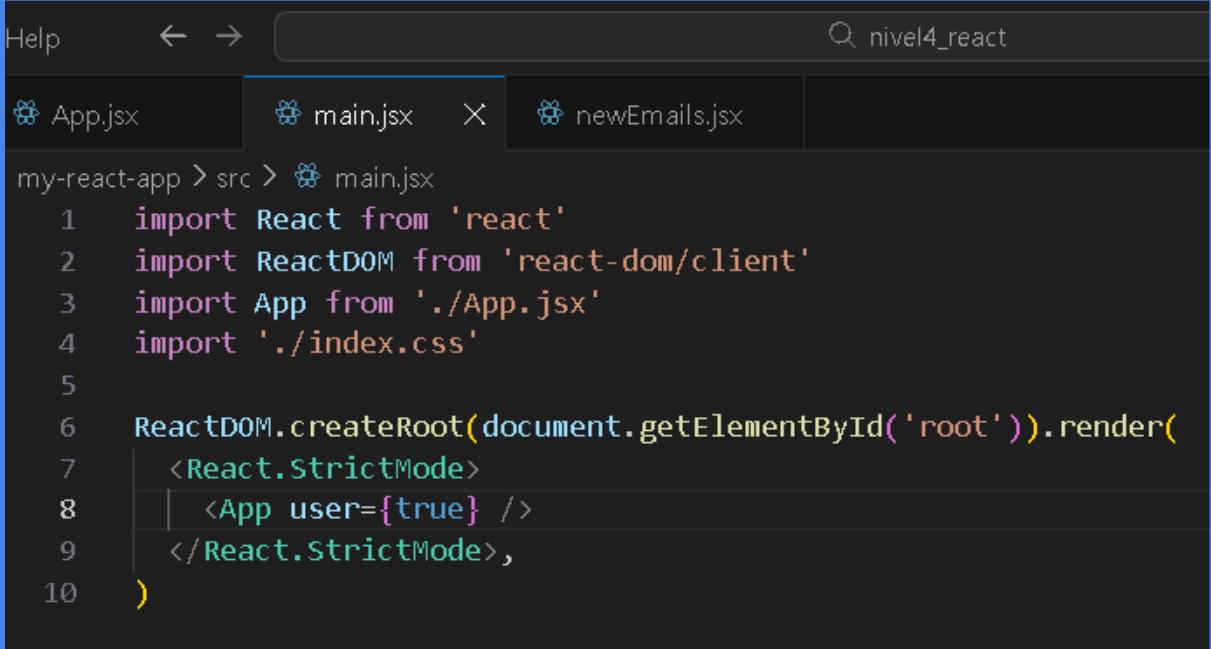
# Nivel 5 React

DAW2-DWEC

## Índice

Parte A: Condicional con if + return.....	2
Parte B: Renderizado parcial usando una variable.....	4
Parte C: Renderizado en línea con el operador &&.....	5
Parte D: Renderizado en línea con operador ternario.....	7
Miniretillo (pero de entrega obligatoria).....	9
Preguntas extra para el PDF.....	10

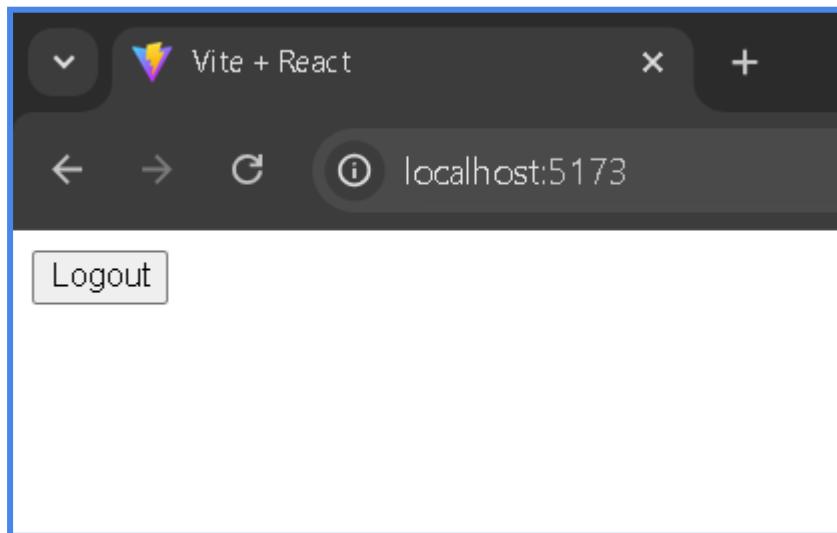
## Parte A: Condicional con if + return



The screenshot shows a code editor window titled "nive14\_react". The tabs at the top are "App.jsx", "main.jsx", and "newEmails.jsx". The "main.jsx" tab is active. The code in "main.jsx" is as follows:

```
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 user={true} />
9   </React.StrictMode>,
10 )
```

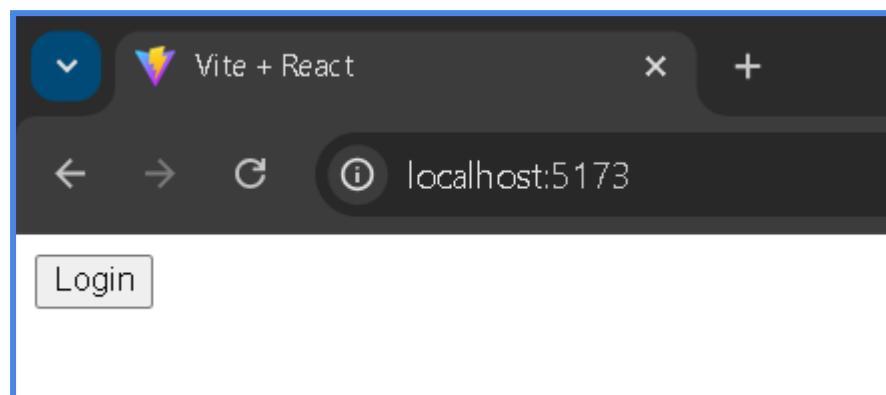
“Figura 01, Condicional con return, user como true, elaboración propia”



“Figura 02, muestra en pantalla, elaboración propia”

```
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 user={false} />
9   </React.StrictMode>,
10 )
```

“Figura 03, user como false, elaboración propia”



“Figura 04, muestra en pantalla, elaboración propia”

## Parte B: Renderizado parcial usando una variable

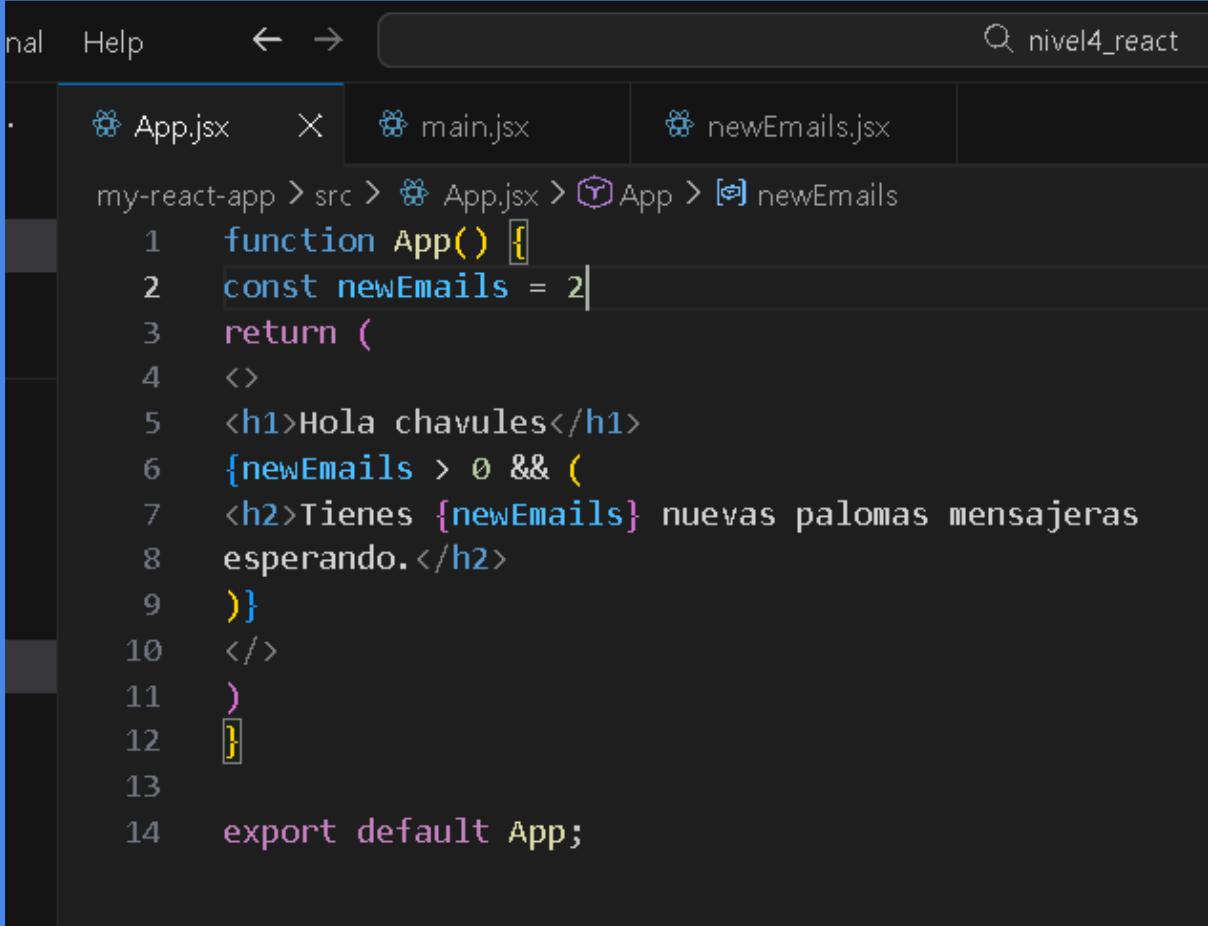


"Figura 05, user como true y frase de esclavillo en h2, elaboración propia"



"Figura 06, user como false y frase de esclavillo en h2, elaboración propia"

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



The screenshot shows a code editor with a dark theme. The file tab at the top has 'App.jsx' selected. The code in 'App.jsx' is as follows:

```
1 function App() {
2   const newEmails = 2
3   return (
4     <>
5     <h1>Hola chavules</h1>
6     {newEmails > 0 && (
7       <h2>Tienes {newEmails} nuevas palomas mensajeras
8       esperando.</h2>
9     )}
10    </>
11  )
12}
13
14 export default App;
```

“Figura 07, variable newEmails igual a 2, elaboración propia”



“Figura 08, muestra en pantalla, elaboración propia”

The screenshot shows a code editor window with a dark theme. At the top, there are tabs for 'App.jsx' (which is currently active), 'main.jsx', and 'newEmails.jsx'. The status bar at the bottom indicates the path: 'my-react-app > src > App.jsx > App > newEmails'. The main area displays the following code:

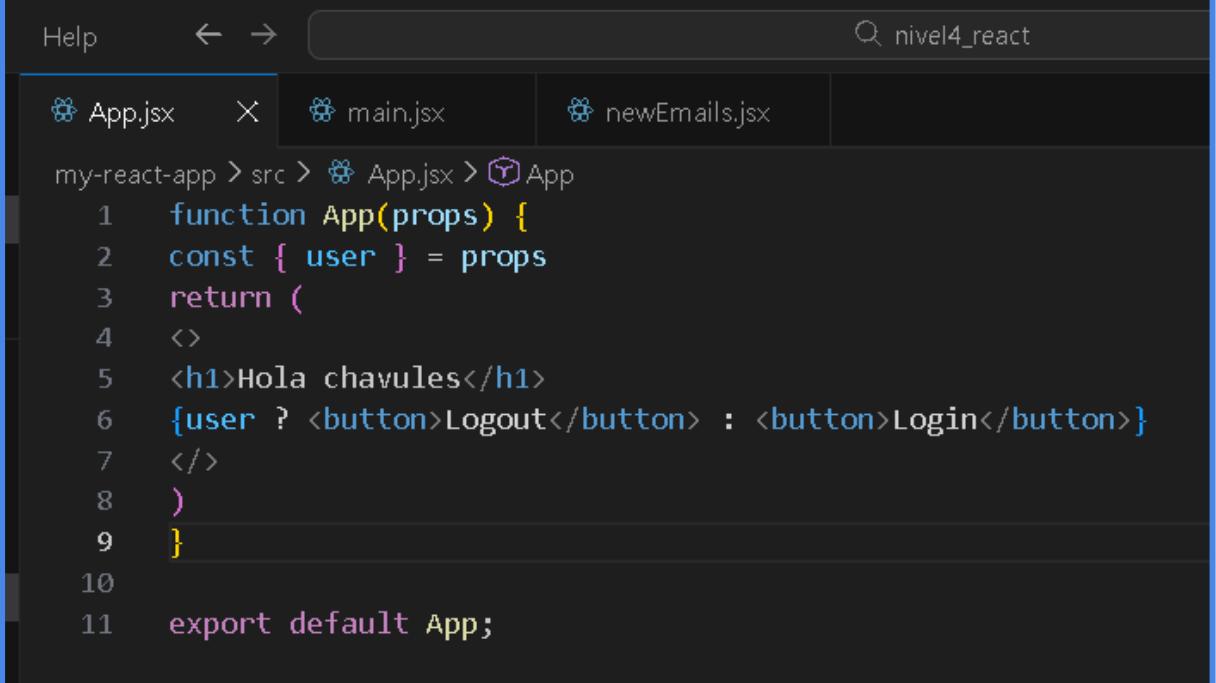
```
1  function App() {
2    const newEmails = 0
3    return (
4      <>
5      <h1>Hola chavules</h1>
6      {newEmails > 0 && (
7        <h2>Tienes {newEmails} nuevas palomas mensajeras
8        esperando.</h2>
9      )}
10     </>
11   )
12 }
13
14 export default App;
```

“Figura 09, variable newEmails igual a 0, elaboración propia”



“Figura 10, muestra en pantalla, elaboración propia”

## Parte D: Renderizado en línea con operador ternario



```
Help ← → 🔍 nivel4_react
App.jsx X main.jsx newEmails.jsx
my-react-app > src > App.jsx > App
1 function App(props) {
2   const { user } = props
3   return (
4     <>
5       <h1>Hola chavules</h1>
6       {user ? <button>Logout</button> : <button>Login</button>}
7     </>
8   )
9 }
10
11 export default App;
```

“Figura 11, operador ternario, elaboración propia”

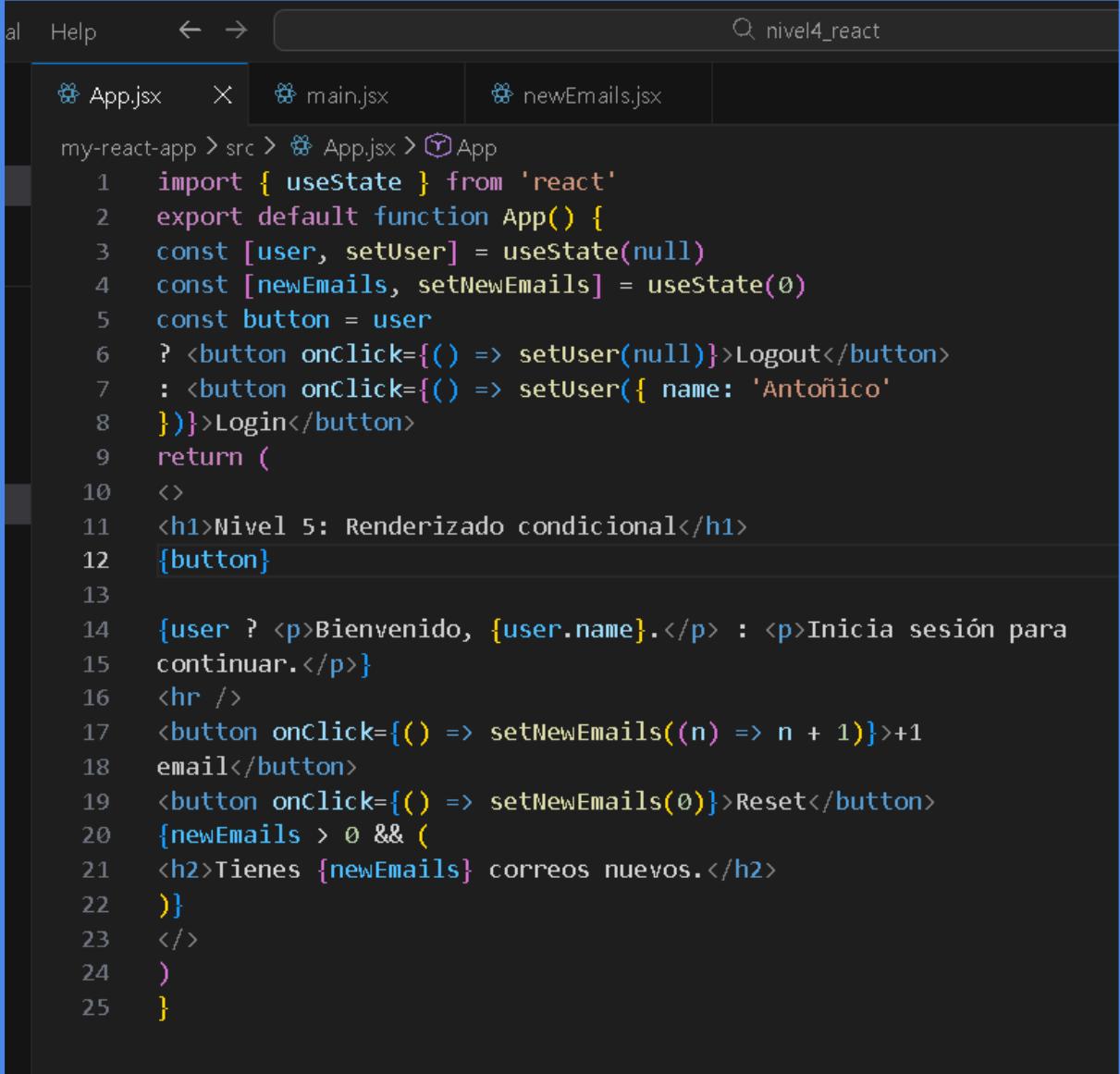


“Figura 12, muestra en pantalla con operador ternario en true, elaboración propia”



"Figura 13, muestra en pantalla con operador ternario en false, elaboración propia"

# Miniretillo (pero de entrega obligatoria)



The screenshot shows a code editor window titled "nivel4\_react". The tab bar has three tabs: "App.jsx" (which is active), "main.jsx", and "newEmails.jsx". The code in "App.jsx" is as follows:

```
my-react-app > src > App.jsx > App
1 import { useState } from 'react'
2 export default function App() {
3   const [user, setUser] = useState(null)
4   const [newEmails, setNewEmails] = useState(0)
5   const button = user
6   ? <button onClick={() => setUser(null)}>Logout</button>
7   : <button onClick={() => setUser({ name: 'Antoñico' })}>Login</button>
8   return (
9     <>
10    <h1>Nivel 5: Renderizado condicional</h1>
11    {button}
12
13
14   {user ? <p>Bienvenido, {user.name}.</p> : <p>Inicia sesión para
15   continuar.</p>}
16   <hr />
17   <button onClick={() => setNewEmails((n) => n + 1)}>+1
18   email</button>
19   <button onClick={() => setNewEmails(0)}>Reset</button>
20   {newEmails > 0 && (
21     <h2>Tienes {newEmails} correos nuevos.</h2>
22   )}
23   </>
24 )
25 }
```

“Figura 14, , elaboración propia”

```
import { useState } from 'react'
export default function App() {
  const [user, setUser] = useState(null)
  const [newEmails, setNewEmails] = useState(0)
```

“Figura 15, useState con estado null, elaboración propia”

```
const button = user
? <button onClick={() => setUser(null)}>Logout</button>
: <button onClick={() => setUser({ name: 'Antoñico'
})}>Login</button>
return (
<>
```

“Figura 16, botón que cambie de login a logout, elaboración propia”

```
{user ? <p>Bienvenido, {user.name}.</p> : <p>Inicia sesión para
continuar.</p>}
<hr />
```

“Figura 17, mensaje de bienvenida con su nombre, elaboración propia”

```
const [user, setUser] = useState(null)
const [newEmails, setNewEmails] = useState(0)
const button = user
```

“Figura 18, Crea un estado newEmails, elaboración propia”

```
<button onClick={() => setNewEmails((n) => n + 1)}>+1
email</button>
<button onClick={() => setNewEmails(0)}>Reset</button>
{newEmails > 0 && (
```

“Figura 19, botón para incrementar newEmails y otro para ponerlo a 0, elaboración propia”

```
const [newEmails, setNewEmails] = useState(0)
const button = user
? <button onClick={-1}>
```

“Figura 20, renderizado parcial, elaboración propia”

## Preguntas extra para el PDF

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

- La diferencia es que con if/return corta la ejecución y lo saca del componente, mientras que el ternario permite elegir entre las 2 opciones sin que se rompa el html

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

- Conviene usarlo cuando solo quiera mostrar algo si se cumple antes la condición o nada si no se cumple, el ternario lo reservaría para cuando tenga que elegir una de las dos opciones si o sí por obligación

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

- Que mantienes el return más limpio y organizado porque así lo ejecuto todo antes y ya después soltar la variable

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

- En casi cualquier panel de usuario se puede usar para que solo se muestren cosas específicas a los administradores y otras cosas distintas a los usuarios normales