

NIVEL 5 REACT



PARTE A: Condicional con if + return.....	2
PARTE B: Renderizado parcial usando una variable.....	3
PARTE C: Renderizado en línea con el comparador && 3	
PARTE D: Renderizado en línea con operador ternario.4	
MINIRETILLO:.....	5
PREGUNTAS EXTRAS.....	6

PARTE A: Condicional con if + return

The screenshot shows a browser window at `localhost:5173` with a green header bar labeled "Centro educativo". Below the header, there is a button labeled "Logout". To the right of the browser is a code editor window titled "App.tsx". The code is as follows:

```
# index.css          App.tsx
src > App.tsx > App
1 // import { useState } from "react"
2
3 function App(): JSX.Element {
4   //const [user, setUser] = useState(true)
5   const user = true
6
7   if (user) {
8     return <button> Logout</button>
9   }
10  return <button>Login</button>
11}
12
13
14 export default App
15
```

The code uses a conditional statement (`if (user)`) to determine whether to render a "Logout" button or a "Login" button. Since `user` is set to `true`, the application displays "Logout".

(Ilustración 1. Captura de pantalla del código y de la aplicación donde se muestra el código siendo la variable “user = true” como valor fijo temporal, y enseñando como la salida en la aplicación es “Logout”. Elaboración propia.)

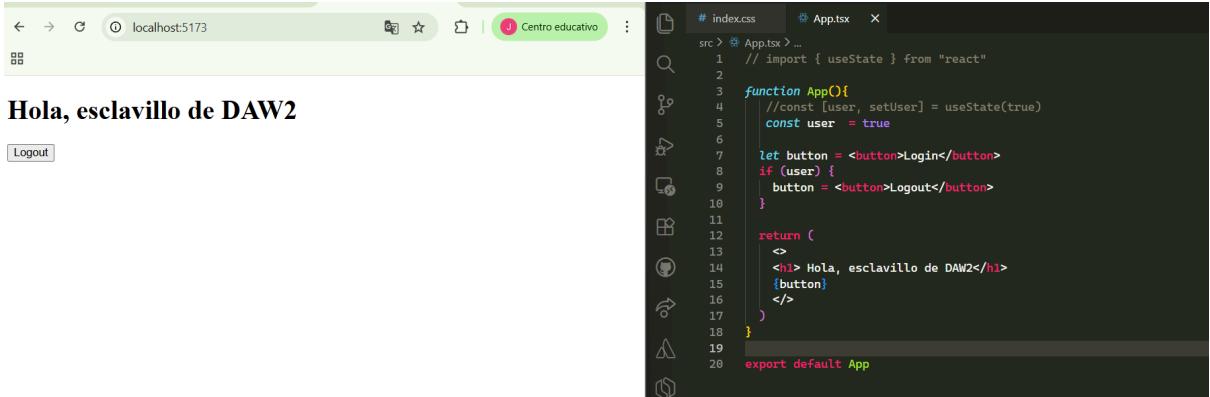
The screenshot shows a browser window at `localhost:5173` with a green header bar labeled "Centro educativo". Below the header, there is a button labeled "Login". To the right of the browser is a code editor window titled "App.tsx". The code is as follows:

```
# index.css          App.tsx
src > App.tsx > App > user
1 // import { useState } from "react"
2
3 function App(): JSX.Element {
4   //const [user, setUser] = useState(true)
5   const user = false
6
7   if (user) {
8     return <button> Logout</button>
9   }
10  return <button>Login</button>
11}
12
13
14 export default App
15
```

The code uses a conditional statement (`if (user)`) to determine whether to render a "Logout" button or a "Login" button. Since `user` is set to `false`, the application displays "Login".

(Ilustración 2. Captura de pantalla del código y la aplicación en el caso contrario. Ponemos el valor de “user” como `false` y nos sale “Login” en la aplicación. Elaboración propia.)

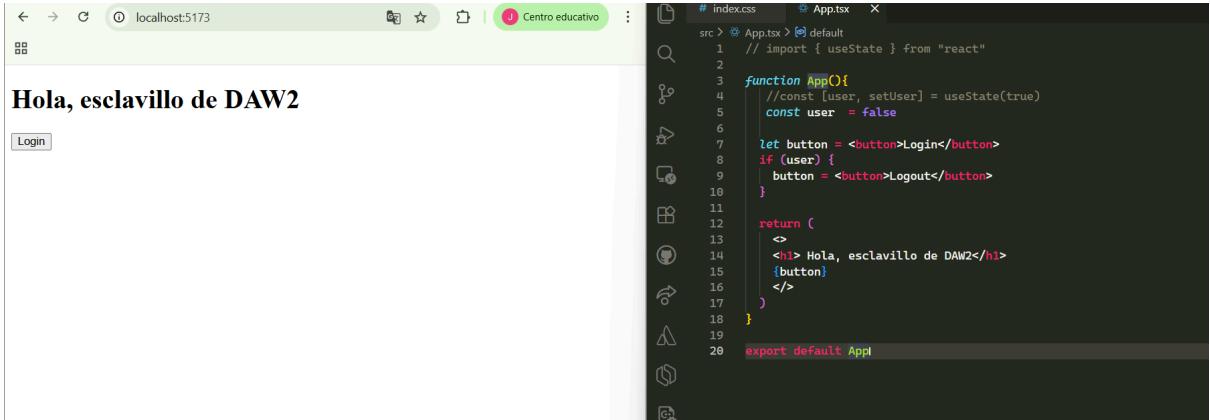
PARTE B: Renderizado parcial usando una variable



The screenshot shows a browser window with the URL `localhost:5173` and a tab titled "Centro educativo". The page content is "Hola, esclavillo de DAW2" with a "Logout" button. To the right is a code editor showing the `App.tsx` file:

```
# index.css  App.tsx
src > App.tsx > ...
1 // import { useState } from "react"
2
3 function App(){
4   //const [user, setUser] = useState(true)
5   const user = true
6
7   let button = <button>Login</button>
8   if (user) {
9     button = <button>Logout</button>
10  }
11
12  return (
13    <>
14      <h1> Hola, esclavillo de DAW2</h1>
15      {button}
16    </>
17  )
18
19
20 export default App
```

(Ilustración 3. Captura de pantalla del código y de la App donde se muestra el botón de “Logout”, declarando el botón como una variable. Elaboración propia.)

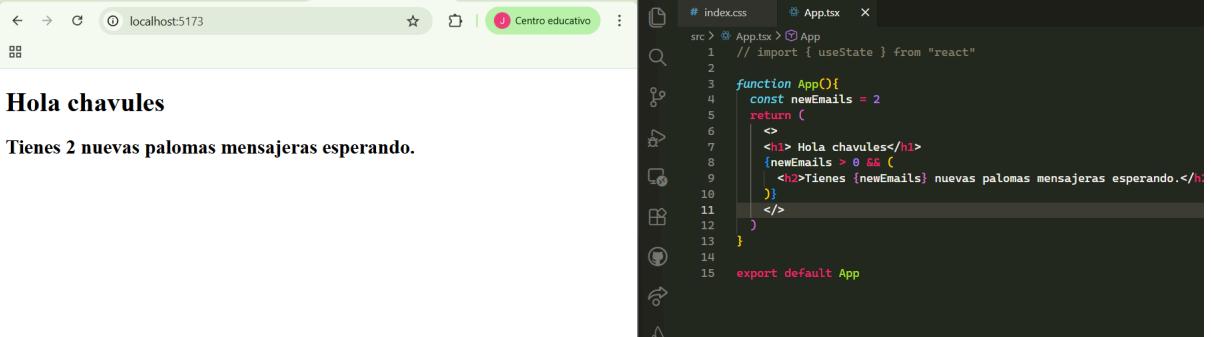


The screenshot shows a browser window with the URL `localhost:5173` and a tab titled "Centro educativo". The page content is "Hola, esclavillo de DAW2" with a "Login" button. To the right is a code editor showing the `App.tsx` file:

```
# index.css  App.tsx
src > App.tsx > default
1 // import { useState } from "react"
2
3 function App(){
4   //const [user, setUser] = useState(true)
5   const user = false
6
7   let button = <button>Login</button>
8   if (user) {
9     button = <button>Logout</button>
10  }
11
12  return (
13    <>
14      <h1> Hola, esclavillo de DAW2</h1>
15      {button}
16    </>
17  )
18
19
20 export default App
```

(Ilustración 4. Captura de pantalla del código y de la App donde se muestra el caso contrario al anterior, cambiamos el valor “`user = false`” para observar el cambio del contenido en la variable “`button`”. Elaboración propia.)

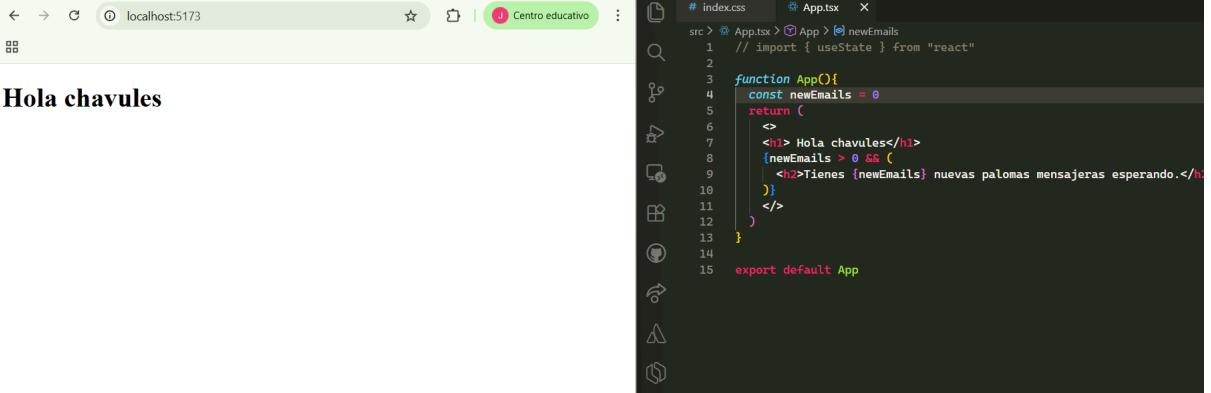
PARTE C: Renderizado en línea con el comparador &&



The screenshot shows a browser window at localhost:5173. The title bar says "Centro educativo". The page content displays "Hola chavules" and "Tienes 2 nuevas palomas mensajeras esperando.". To the right is a code editor with App.tsx:

```
# index.css ② App.tsx x
src > ③ App.tsx > App
1 // import { useState } from "react"
2
3 function App(){
4   const newEmails = 2
5   return (
6     <>
7       <h1> Hola chavules</h1>
8       {newEmails > 0 && (
9         <h2>Tienes {newEmails} nuevas palomas mensajeras esperando.</h2>
10      )}
11    </>
12  )
13}
14
15 export default App
```

(Ilustración 5. Captura de pantalla del código y de la App donde creamos la variable “newEmails”. Al ser mayor que 2, nos muestra el mensaje, haciendo uso de &&. Elaboración propia. Elaboración propia.)

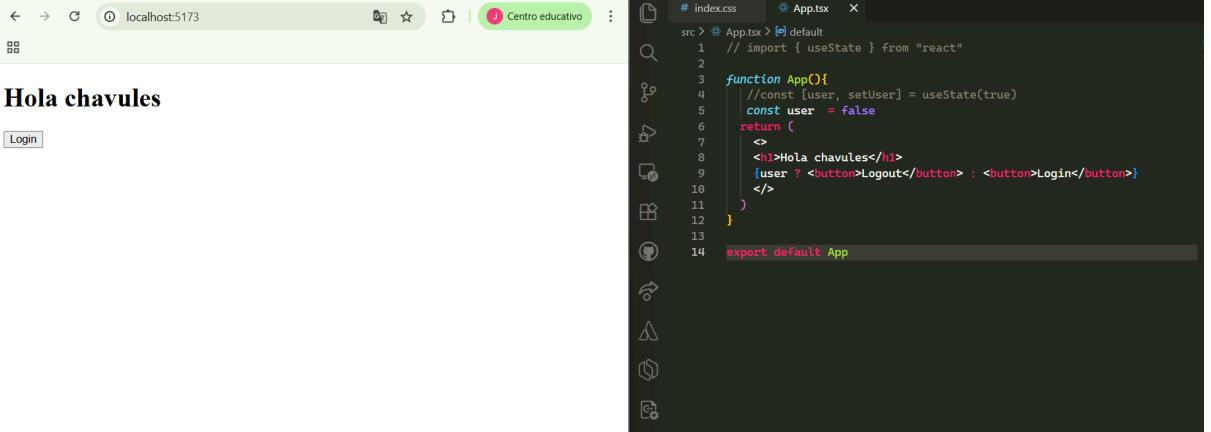


The screenshot shows a browser window at localhost:5173. The title bar says "Centro educativo". The page content displays "Hola chavules". To the right is a code editor with App.tsx:

```
# index.css ② App.tsx x
src > ③ App.tsx > App > newEmails
1 // import { useState } from "react"
2
3 function App(){
4   const newEmails = 0
5   return (
6     <>
7       <h1> Hola chavules</h1>
8       {newEmails > 0 && (
9         <h2>Tienes {newEmails} nuevas palomas mensajeras esperando.</h2>
10      )}
11    </>
12  )
13}
14
15 export default App
```

(Ilustración 6. Captura de pantalla del código y de la App, donde mostramos el caso de que “newEmails” sea 0, no mostrándose el mensaje en la App. Elaboración propia.)

PARTE D: Renderizado en línea con operador ternario

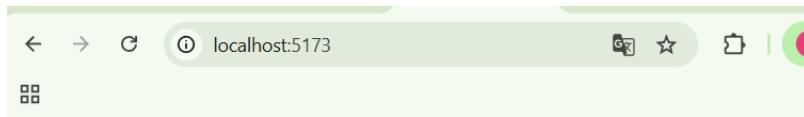


The screenshot shows a browser window at localhost:5173. The title bar says "Centro educativo". The page content displays "Hola chavules" and a "Login" button. To the right is a code editor with App.tsx:

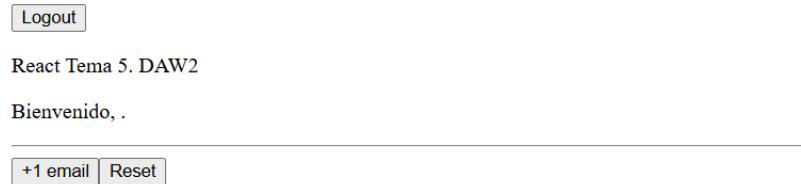
```
# index.css ② App.tsx x
src > ③ App.tsx > default
1 // import { useState } from "react"
2
3 function App(){
4   //const [user, setUser] = useState(true)
5   const user = false
6   return (
7     <>
8       <h1>Hola chavules</h1>
9       {user ? <button>Logout</button> : <button>Login</button>}
10     </>
11   )
12}
13
14 export default App
```

(Ilustración 7. Captura de pantalla del código y de la App, donde reescribimos el botón de Login/Logout usando el operador ternario. Elaboración propia.)

MINIRETILLO:



Nivel 5: Renderizado condicional



(Ilustración 8. Captura de pantalla de la App corriendo con todo lo que se pide en el ejercicio. Todas las funcionalidades implementadas y funcionando a la perfección. Elaboración propia.)

```
# index.css  App.tsx 2 ×
src > App.tsx > App
1 import { useState } from 'react';
2
3 export default function App() {
4   const [user, setUser] = useState(true);
5   const [newEmails, setNewEmails] = useState(0);
6
7   const button = user
8     ? <button onClick={() => setUser(!user)}>Logout</button>
9     : <button onClick={() => setUser({ name: 'Antoñico' })}>Login</button>;
10
11   return [
12     <>
13       <h1>Nivel 5: Renderizado condicional</h1>
14       <button>
15
16         <p>React Tema 5. DAW2</p>
17
18         {user ?
19           <p>Bienvenido, {user.name}.</p>
20         } :
21           <p>Inicia sesión para continuar.</p>
22         )
23
24       <hr />
25
26       <button onClick={() => setNewEmails((n) => n + 1)}>+1 email</button>
27       <button onClick={() => setNewEmails(0)}>Reset</button>
28
29       {newEmails > 0 && (
30         <h2>Tienes {newEmails} correos nuevos.</h2>
31       )}
32     </>
33   ];
34 }
35
```

(Ilustración 9. Captura de pantalla del código donde mostramos las implementaciones conforme se van pidiendo en el ejercicio. Usamos el useState, el botón cambia según su valor, mostramos el mensaje si esta logueado, creamos un estado newEmails y mostramos si newEmails > 0, incluye botón para incrementar newEmails o igualarlo a 0, usamos al menos un ternario... Elaboración propia.)

PREGUNTAS EXTRAS

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

- If/return: Sirve para decidir qué componente completo devolver antes del return principal.
- Ternario se usa dentro del JSX, para cambiar cosas inline sin salir del return

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

Cuándo quieres renderizar algo solo si la condición es true, sin necesidad de un "else".

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

Permite tener lógica más compleja o limpia antes del JSX, y evita ternarios gigantes adentro del return

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

Un ejemplo sería el e-commerce donde se muestra un carrito de compras vacío (con un mensaje y un botón de "Seguir comprando") si el usuario no tiene productos, pero se muestra la lista detallada de productos, precios y checkout si el carrito tiene elementos, controlando la visibilidad de estas secciones dinámicamente según el estado del carrito, lo cual es fundamental para la lógica de la aplicación y la experiencia de usuario.