

## **Índice**

[Parte A: JSX básico \(guardar y devolver un elemento\):](#)

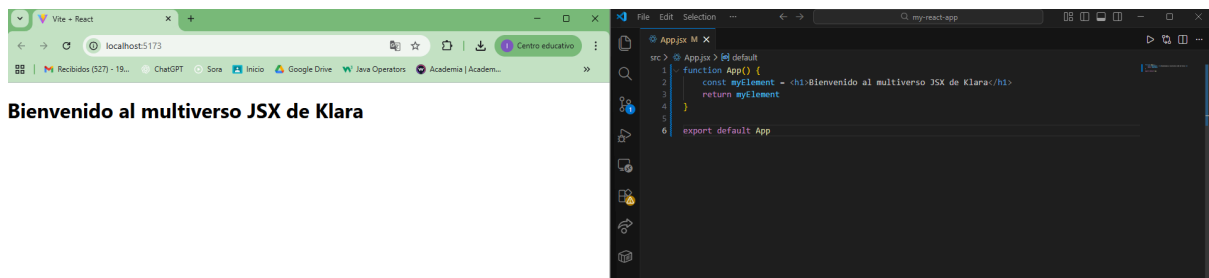
[Parte B: Expresiones dentro de JSX con { }:](#)

[Parte C: Renderizar una lista con map\(\) + key:](#)

[Parte D: className y estilos rápidos:](#)

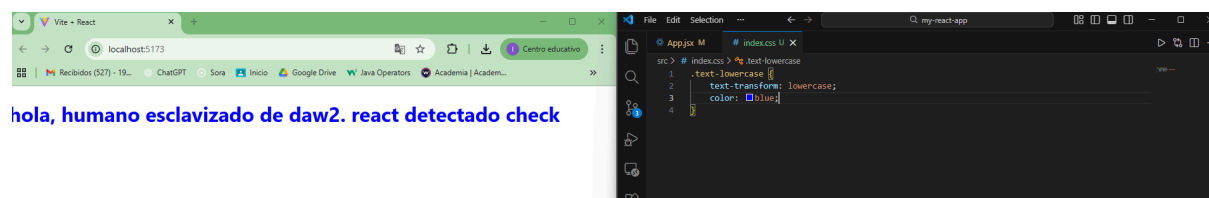
[Preguntas](#)

## Parte A: JSX básico (guardar y devolver un elemento):



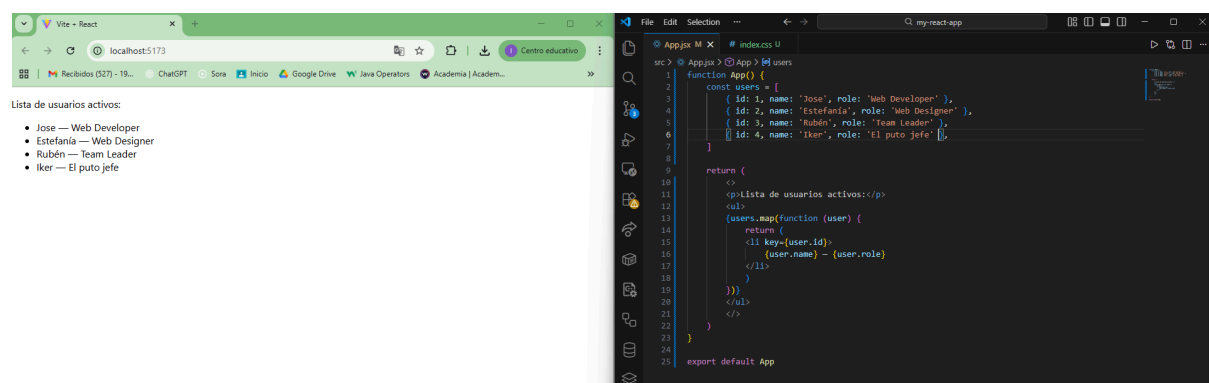
(Captura de pantalla mostrando el código y su funcionamiento a la vez, elaboración propia)

## Parte B: Expresiones dentro de JSX con {}:



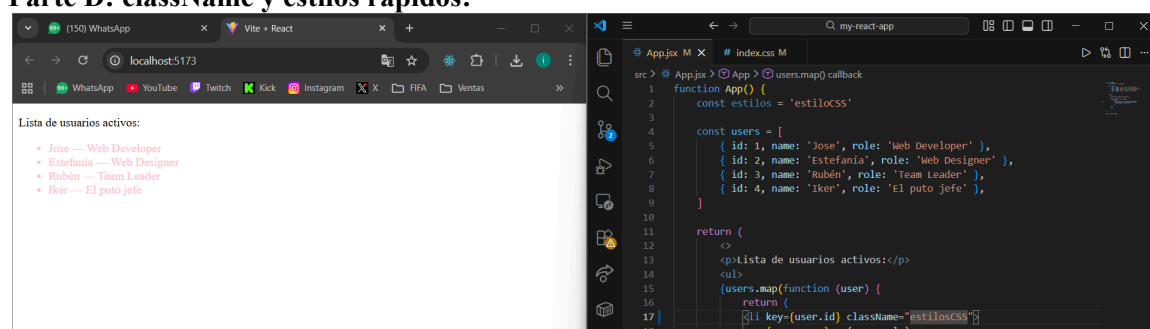
(Captura de pantalla mostrando el código y su funcionamiento a la vez, elaboración propia)

## Parte C: Renderizar una lista con map() + key:



(Captura de pantalla mostrando el código y su funcionamiento a la vez, elaboración propia)

## Parte D: className y estilos rápidos:



(Captura de pantalla mostrando el código y su funcionamiento a la vez, elaboración propia)

## **Preguntas**

### **1. ¿Qué es JSX?**

JSX es una forma de escribir en React usando una sintaxis parecida a HTML dentro de JavaScript. Sirve para definir cómo se ve la interfaz de la aplicación de forma clara y visual.

### **2. ¿Por qué usamos { } en JSX?**

Se usan para insertar código JavaScript dentro del HTML de React.  
Por ejemplo: `<h1>Hola {nombre}</h1>` muestra el valor de una variable.

### **3. ¿Para qué sirve la prop key?**

Sirve para identificar de forma única cada elemento de una lista.  
Así React sabe qué elementos cambian y puede actualizar la interfaz correctamente.

### **4. ¿Por qué usamos className y no class?**

Porque class es una palabra reservada en JavaScript.  
React usa className para poder aplicar estilos CSS sin causar conflictos.