

# Documento de estudio para React y TailWindCSS

Documento hecho por **José Cifuentes** para el proyecto “MedManager”



## ¿Qué es React?

React es una biblioteca de Javascript que fue creada por Meta (Que era Facebook) para construir interfaces de usuario, centrándose en aplicaciones web del lado frontend.

La idea principal de react es que la pagina o aplicación se pueda dividir en componentes reutilizables, como botones, menús, formularios, y otras cosas más. Cada componente se actualiza de forma eficiente cuando cambian los datos, sin necesidad de recargar toda la pagina (Ideal para cuando cambia el estado de una licencia, por ejemplo, si es revisada se avisa con un numerito en la barra de estado que se pone encima de “revisado” y ese numero muestra cuantas licencias hay con ese estado, un ejemplo sería el botón de notificaciones de Facebook, que muestra un numero que dice cuantas notificaciones tienes y cambia en tiempo real).

Una de las características claves de react es

- Usa una sintaxis llamada JSX, que mezcla HTML y Javascript
- Trabaja con un DOM virtual (Documento de modelo de objetos) para hacer más rápidas las actualizaciones\*
- Facilita crear interfaces dinámicas y creativas
- Cuenta con otras herramientas como React Router para navegación o Redux para manejo de estados

## \*¿Qué es un DOM Virtual?:

Un DOM (Document Object Model) es la representación de la página web en memoria, básicamente un árbol con todos los elementos (<div>, <p>, <button>, etc.).

Modificar el DOM real directamente es lento, porque el navegador tiene que recalcular estilos, posiciones y redibujar la página.

Lo que hace react con el DOM Virtual es que crea una copia ligera en la memoria, por eso se llama DOM "Virtual". Cuando algo cambia, react actualiza el DOM Virtual y luego lo compara con el DOM Virtual anterior para detectar que se cambió exactamente, para después aplicar los cambios en el DOM real.

### **¿Por qué utilizar React y no CSS Pelado o Bootstrap?**

Al ser react pensado para manejar datos en tiempo real, cuenta con mejor soporte a ello ya que el manejo se hace de manera mas eficiente. Como nuestra página tendrá datos constantemente siendo manejados, darle al usuario una experiencia mejor en tiempo real es lo que buscamos.

Mientras que CSS Pelado y Bootstrap se centran más en apariencia o personalización. No al manejo de datos a tiempo real

### **Requisitos Minimos para usar React:**

- Conocimientos: HTML, CSS y Javascript moderno (ES6)
- Entorno: Node.js, npm instalados (Esto es para gestionar las dependencias) y un editor de código, como Visual Studio Code o Notepad++

Para la inicialización de un proyecto con react hay dos maneras. (Escritas en un editor de texto)

1. **create-react-app**
2. **Vite**

Ambas hacen lo mismo, solo que una explica que es lo que hace mientras que la otra es más rápida de escribir y ejecutar ya que es más corta.

### **Como usar React apropiadamente:**

- Hay que dividir la interfaz en compoentes (ej: botón, navbar, formulario, etc)
- Esos componentes se tienen que escribir en JSX (mescla de HTML y JS).
- Hacer que react renderize los componentes en el navegador para verificar si el estado de ellos se actualiza automáticamente al cambiar.

Ejemplo simple escrito en JSX:

```
function Saludo({ nombre }) {  
  return <h1>Hola, {nombre}!</h1>;  
}
```

La **primera línea** define un componente llamado “saludo”.

Recibe un prop (propiedad) llamado “nombre”.

{ nombre } es desestructuración, que equivale a “props.nombre”.

La **segunda línea** devuelve un elemento JSX y renderiza un <h1> con el texto “Hola, ...” y coloca el valor de nombre.

Si tuviéramos que usar ese ejemplo en un caso real para mostrar un nombre, se usa el siguiente código:

```
<Saludo nombre="Jose" />
```

Y lo que se vería en la página sería esto:

```
Hola, Jose!
```

Para entender mejor cómo funciona React he incluido unos links y videos de apoyo que estarán bajo este texto, son bastante bien explicados y sirven de mucho.

**Enlaces de React,**

- <https://es.react.dev/learn>, (Aprendizaje con página oficial)
- [https://youtu.be/bvxm389cYVI?si=48\\_ADOXPTY9JxIB](https://youtu.be/bvxm389cYVI?si=48_ADOXPTY9JxIB), (Explicación Animada)

**Enlaces de React Router,**

- <https://youtu.be/JNhhdKcuyog?si=somW29rG7RCc8Jlj>, (Aprendizaje rápido)
- <https://youtu.be/xE7TBCR6cj0?si=7kXCnuVAULwM4ZH8> (Añadir rutas a app)

## ¿Qué es TailwindCSS?

Es un framework de CSS utilitario.

En lugar de usar clases prearmadas (como en Bootstrap) o escribir CSS desde cero, te da pequeñas clases utilitarias (por ejemplo: text-center, bg-blue-500, p-4) que aplicas directamente en el HTML.

Con esas clases vas componiendo estilos sin tener que crear archivos CSS enormes.

La idea es rapidez y consistencia: no inventas nombres de clases, usas utilidades ya definidas.

¿Cómo se usa?

Instalación mínima:

Necesitas Node.js y npm. Al igual que React

Instalas Tailwind con:

```
npm install -D tailwindcss
```

```
npx tailwindcss init
```

Esto genera los archivos tailwind.config.js y postcss.config.js.