

# Introducción a React - Paso a paso desde lo que ya sabemos

---

## ¿Por qué estamos acá?

Hoy vamos a dar el salto desde la forma tradicional de construir interfaces (maquetado + JavaScript puro) hacia **una nueva forma de pensar las interfaces**: la **programación declarativa** y basada en **componentes reutilizables**. Eso es exactamente lo que nos propone **React**.

Vamos a partir de lo que ya hicimos juntos: una maqueta estática de nuestro sistema **BiciAlquileres**, armada con HTML, CSS y un poquito de JS.

Y desde ahí, vamos a **evolucionar progresivamente**:

1. Maquetado → JS puro generando HTML
2. React sin JSX (usando `React.createElement`)
3. React con JSX
4. React con Vite
5. React con estructura por componentes, CSS modular y Bootstrap

---

## Punto de partida - HTML generado desde JavaScript

Así como imprimíamos en consola en los primeros ejercicios, también podemos generar HTML dinámicamente desde código JS:

```
<!-- index.html -->
<body>
  <div id="root"></div>
  <script src="app.js"></script>
</body>
```

```
// app.js
const titulo = document.createElement("h1");
titulo.innerText = "🚀 Bienvenidos al mundo de la programación web";

const contenedor = document.getElementById("root");
contenedor.appendChild(titulo);
```

## Problemas de este enfoque

- Código muy verboso
- Difícil de leer y mantener
- No hay separación de estructura y lógica

- Difícil reusar y probar

---

## ¿Qué es React?

React es una **librería de JavaScript para construir interfaces de usuario**. Fue creada por Facebook en 2013 y está pensada para:

- Crear componentes reutilizables
- Trabajar de forma declarativa
- Optimizar las actualizaciones de la interfaz
- Mejorar la legibilidad y mantenibilidad del código

En lugar de decirle al navegador *cómo* cambiar cada parte del DOM, le decimos *qué* queremos mostrar, y React se encarga del resto.

Es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre. En el proyecto hay más de mil desarrolladores libres.

 Documentación oficial: <https://react.dev/>

---

## Primer contacto: React sin JSX

React transforma el código que escribimos en objetos JS que representan el DOM. Podemos escribir directamente:

```
const titulo = React.createElement("h1", null, "Hola desde React sin JSX");

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(titulo);
```

Esto funciona, pero sigue siendo poco legible.


---

## JSX - Escribir código más parecido a HTML

JSX es una **extensión de JavaScript** que nos permite escribir estructura similar a HTML dentro del JS:

```
const titulo = <h1>Hola desde JSX</h1>;

root.render(titulo);
```

 Pero... JSX no es HTML ni JS puro

- El navegador no lo entiende directamente
- Necesitamos un transpilador (como **Babel**) que lo convierta en `React.createElement(...)`

- Esto lo maneja automáticamente Vite por debajo

🔗 Más sobre JSX: <https://react.dev/learn/writing-markup-with-jsx>

---

## ⚙️ Crear un proyecto con Vite

Para trabajar de manera moderna, usamos **Vite**. Es un *bundler* rápido y liviano que reemplaza al viejo `create-react-app`.

```
npm create vite@latest
```

Elegimos `React` como framework y `JavaScript` como lenguaje.

Vite nos da una estructura limpia, rápida de cargar y lista para usar con JSX y CSS modular.

---

## 🧩 Componentes: pensar la interfaz como bloques

En lugar de pensar la app como una sola página, la separamos en **componentes reutilizables**:

```
// Encabezado.jsx
function Encabezado() {
  return <header className="bg-dark text-white">BiciAlquileres</header>;
}
```

```
// App.jsx
import Encabezado from './components/Encabezado';

function App() {
  return (
    <>
      <Encabezado />
      <Inicio />
      <PiePagina />
    </>
  );
}
```

Esta es la base de cualquier app React: pequeños componentes reutilizables que se combinan.

---

## 🎨 Estilos en React

- ◇ CSS global

```
import './assets/css/bicialquileres.css';
```

### ◇ CSS por componente

```
import './Inicio.css';
```

---

## 🔗 Resultado final del paso 5

- Página **Inicio** completamente en React
- Cards con Bootstrap
- Botón de acción
- Navbar con pills estáticos (**href="#"**)
- Footer con **<footer>** semántico
- Código legible, organizado y mantenible

---

## 🚴 ¿Y después?

En las próximas clases vamos a ir incorporando:

- 📁 Props
- 🧠 **useState**
- 📦 Navegación con **react-router-dom**
- 🔌 Conexión con backend (más adelante)

---

## 📄 Enlaces útiles

- 🧠 JSX vs HTML: <https://react.dev/learn/writing-markup-with-jsx>
- 📦 Vite + React: <https://vitejs.dev/guide/>
- 🍷 Bootstrap Icons: <https://icons.getbootstrap.com/>
- 🧱 Pensar en componentes: <https://react.dev/learn/thinking-in-react>

---

## 🧠 Reflexión final

React no es solo una herramienta: es una nueva forma de pensar la interfaz.  
No decimos “hacé esto paso a paso”, sino: **“esto es lo que quiero ver”**.

---

¡Vamos a ver todo esto en acción! Abrimos el proyecto, recorremos el código, y después nos metemos en los componentes 🙌 🤖