

Guía Express

```
# Crear Proyecto React con Vite y Limpiar con script clean_react.sh
# Guia Estructura React
```

Crear Proyecto React

1. Crear proyecto

```
npm create vite@latest mi-proyecto-react -- --template react
cd mi-proyecto-react
npm install
npm run dev
```

- `npm create vite@latest ...` → Genera la estructura base.
- `cd mi-proyecto-react` → Entrás en tu carpeta.
- `npm install` → Instala dependencias.
- `npm run dev` → Levanta servidor local → `http://localhost:5173`

2. Estructura básica

```
/src
├─ App.jsx      # Componente principal
├─ main.jsx     # Entrada React
└─ components/  # Aquí pondrás tus piezas
```

3. Ver en navegador

- Abre `http://localhost:5173`
- Verás la plantilla de Vite + React funcionando.
- Edita `App.jsx` para ver el hot reload en acción.

Limpiar con script

4. Limpiar proyecto con :

```
~/scripts/clean_react.sh
Desde la ruta del proyecto en sí, ejecuta esta script.
```

Tip:
Piensa en cada componente como un bloque Lego .
Crea piezas pequeñas → luego encájalas.

ESTRUCTURA BASE REACT + useState + Props

1. import { useState } from 'react';

```
**Analogía:** "Traigo la caja mágica desde la fábrica React."  
**Regla:** Siempre que quieras guardar datos que cambian, usa `useState`.
```

2. export default function Nombre({ prop }) { ... }

```
**Analogía:** "Creo un molde LEGO llamado Nombre que acepta una etiqueta extra (`prop`)."  
**Regla:** Los componentes siempre empiezan en mayúscula y devuelven JSX.
```

3. const [estado, setEstado] = useState(0);

```
**Analogía:** "Creo una caja `estado` que empieza en 0. La llave `setEstado` la abre y cambia."  
**Regla:** `[ ]` es "caja doble": valor actual y llave.
```

4. return (<div>...</div>)

```
**Analogía:** "Todo lo que devuelve la función es la fachada de la casa."  
**Regla:** Siempre devuelve UN elemento padre.
```

5. {} dentro de JSX

```
**Analogía:** "Ventana mágica para meter JS dentro del HTML."  
**Regla:** Todo dentro de `{}` es JS puro.
```

6. Función flecha ()=>

****Analogía:**** "El arquero dispara una acción."
****Regla:**** `()=>` ejecuta algo al hacer clic u otro evento.

```
<button onClick={() => setEstado(estado + prop)}>Acción</button>
```

7. Props { prop }

****Analogía:**** "Etiqueta que dice: súbeme la escalera de 2 en 2."
****Regla:**** Prop es el dato dinámico que pasa a tu molde.

```
<ContadorPaso paso={2} />
```

REGLAS RÁPIDAS

Símbolo	Significado	Regla
{ }	Caja mágica JS	"Abro JS"
()	Manos que agrupan	"Agrupo args o JSX"
=>	Flecha	"Acción disparada"
[]	Caja doble	"Valor + llave para abrir"
useState()	Caja mágica	"Guardo datos reactivos"
prop	Etiqueta	"Dato dinámico para el molde"

EJERCICIO EXPRESS

Escribe **la** plantilla 5 veces a mano.
Cambia nombre, **prop** y valor inicial.
Léelo **en** voz alta: "Traigo caja, creo molde, devuelvo fachada con JS adentro."

MANTRA FINAL

"React es Lego:**** `import` traigo piezas, `useState` creo caja, `{}` meto JS, `()=>` disparo acción."