

## GUÍA PRÁCTICA: PRIMEROS PASOS CON REACT

**Objetivo:** Aprender los conceptos básicos de React creando una SPA (Single Page Application) simple con navegación y un formulario interactivo.

### Requisitos Previos

- Tener instalado **Node.js**  
Descárgalo desde: <https://nodejs.org>
- Tener instalado un editor de código como **Visual Studio Code**  
<https://code.visualstudio.com/>

### Paso 1: Crear un nuevo proyecto React con Vite

1. Abre la terminal o consola.
2. Escribe los siguientes comandos:

```
npm create vite@latest my-app -- --template react
```

```
cd my-app
```

```
npm install
```

```
npm run dev
```

Esto generará un proyecto con React + Vite, instalará dependencias y abrirá la app en el navegador. Se abrirá tu app en <http://localhost:5173>.

### Estructura Básica del Proyecto

- `src/` → Carpeta donde trabajarás.
- `src/App.jsx` → Archivo principal.
- `src/pages/` → Carpeta donde crearemos nuestras páginas.
- `src/components/` → Carpeta para componentes reutilizables.

### Paso 2: Crear componentes y usar props

#### Componente funcional con TypeScript:

```
// src/components/Titulo.tsx
```

```
type Props = {
```

```
  texto: string;
```

```
};
```

```
export default function Titulo({ texto }: Props) {  
  return <h1>{texto}</h1>;  
}
```

### Usar el componente en App:

```
// src/App.tsx  
  
import Titulo from './components/Titulo';  
  
function App() {  
  return (  
    <div>  
      <Titulo texto="Bienvenido a React + TypeScript" />  
    </div>  
  );  
}
```

```
export default App;
```

### Paso 3: Navegación con React Router

1. Instala React Router:

```
npm install react-router-dom
```

2. Crea páginas:

```
// src/pages/Inicio.tsx  
  
export default function Inicio() {  
  return <h2>Página de Inicio</h2>;  
}
```

```
// src/pages/Contacto.tsx  
  
export default function Contacto() {  
  return <h2>Página de Contacto</h2>;  
}
```

3. Configura rutas en App.tsx:

```
import { BrowserRouter, Routes, Route, Link } from 'react-router-dom';
```

```

import Inicio from './pages/Inicio';
import Contacto from './pages/Contacto';

function App() {
  return (
    <BrowserRouter>
      <nav>
        <Link to="/">Inicio</Link> | <Link to="/contacto">Contacto</Link>
      </nav>
      <Routes>
        <Route path="/" element={<Inicio />} />
        <Route path="/contacto" element={<Contacto />} />
      </Routes>
    </BrowserRouter>
  );
}

```

```
export default App;
```

Ya puedes navegar sin recargar la página (SPA).

#### **Paso 4: Formulario con useState y eventos**

```
// src/pages/Contacto.tsx
```

```

import { useState, FormEvent } from 'react';

export default function Contacto() {
  const [nombre, setNombre] = useState("");
  const [mensaje, setMensaje] = useState("");
  const [enviado, setEnviado] = useState(false);

  const manejarEnvio = (e: FormEvent<HTMLFormElement>) => {
    e.preventDefault();
    console.log(`Mensaje enviado por ${nombre}: ${mensaje}`);
  };
}

```

```

    setEnviado(true);

    setNombre("");

    setMensaje("");
};

return (
  <div>

    <h2>Formulario de Contacto</h2>

    {enviado && <p style={{ color: 'green' }}>¡Mensaje enviado!</p>}

    <form onSubmit={manejarEnvio}>
      <div>
        <label>Nombre:</label><br />
        <input
          type="text"
          value={nombre}
          onChange={(e) => setNombre(e.target.value)}
          required
        />
      </div>
      <div>
        <label>Mensaje:</label><br />
        <textarea
          value={mensaje}
          onChange={(e) => setMensaje(e.target.value)}
          required
        />
      </div>
      <button type="submit">Enviar</button>
    </form>
  </div>
);

```

```
</div>

);

}
```

### Resultado esperado

Tendrás una aplicación con:

- Navegación entre "Inicio" y "Contacto"
- Un formulario interactivo que muestra un mensaje al enviar

### Glosario de conceptos clave

Concepto	Explicación breve
SPA	Aplicación de una sola página (sin recargas)
Componente	Función que devuelve JSX (estructura HTML)
Props	Parámetros que se pasan a un componente
State	Variables internas del componente (con useState)
useState	Hook que crea y actualiza estados
Evento	Acción del usuario (click, input, submit, etc.)
Routing	Navegación entre páginas sin recarga
TypeScript	Superset de JavaScript con tipado estático

### Actividad Final

**Objetivo:** Aplicar lo aprendido para crear una SPA personalizada.

#### Instrucciones:

1. Crea una nueva página llamada **"Acerca de"**.
2. Agrega navegación a esta página desde el menú principal.
3. Añade un formulario con los siguientes campos:
  - Nombre completo
  - Correo electrónico
  - Mensaje
4. Al enviar, muestra un mensaje personalizado:  
*"Gracias, [nombre]. Te responderemos a [correo]."*
5. Usa useState para manejar el estado de cada campo.

6. Valida que ningún campo esté vacío.