



# ReactJs

---

## 1. Introducción a las herramientas

- **ReactJS:** Es una librería de JavaScript para construir interfaces de usuario. Utiliza componentes para estructurar las aplicaciones.
  - **npm:** Es el gestor de paquetes de Node.js. Se usa para instalar dependencias en tu proyecto.
  - **Node.js** es un entorno de ejecución de JavaScript en el servidor que permite ejecutar código JavaScript fuera del navegador, facilitando la creación de aplicaciones backend y el uso de herramientas como npm.
  - **Vite:** Es una herramienta de construcción que es muy rápida y optimizada para proyectos de JavaScript, especialmente React. A diferencia de otros bundlers, Vite hace un arranque muy rápido gracias a que usa ESM Modules.
  - **Axios:** Es una librería para hacer peticiones HTTP (comúnmente se usa para interactuar con APIs REST).
  - **JSON Server:** Es una herramienta para crear un servidor REST API de manera rápida y sencilla, lo que te permite simular un backend mientras desarrollas tu aplicación.
- 
- Para ejecutar ReactJs se necesita nodeJs, el cual se instala con nvm. El gestor de paquetes para las librerías es npm y para algunos comandos como JsonServer se usa npx.
    - NodeJs
      - nvm.
    - npm
      - npx.

## 2. Instalación de las herramientas necesarias

### Paso 1: Instalación de Node.js y npm

```
node -v # sudo apt install nodejs
npm -v # sudo apt install npm
#Esos comandos están anticuados, mejor hacerlo con nvm.
nvm -v
npm -v
```

### Paso 2: Crear un nuevo proyecto con Vite

Vite es una excelente herramienta para iniciar proyectos en React. Para crear un nuevo proyecto:

- Abre la terminal y navega hasta la carpeta donde quieras crear el proyecto.
- Ejecuta el siguiente comando para crear un nuevo proyecto React usando Vite:

```
npm create vite@latest
```

- Una vez que se cree el proyecto, ingresa al directorio del proyecto:

```
cd nombreProyecto
```

- Instala las dependencias necesarias:

```
npm install
```

Esto instalará todas las dependencias necesarias para tu proyecto, incluyendo React y ReactDOM.

Se crea node\_modules (están los archivos de las librerías) y package-lock.json. Además package.json (archivo de configuración con las dependencias) es super importante, sin él se rompe todo.

### Paso 3: Iniciar el servidor de desarrollo

Para iniciar el proyecto y ver la aplicación en el navegador, usa el siguiente comando:

```
npm run dev
```

Esto debería abrir tu navegador en `http://localhost:5173`, donde podrás ver tu nueva aplicación React en funcionamiento.

### 3. Instalar Axios y JSON Server

#### Paso 1: Instalar Axios

Axios es una librería que se usa para hacer peticiones HTTP. Para instalar Axios en tu proyecto, ejecuta:

```
npm install axios
```

#### Paso 2: Instalar JSON Server

JSON Server simula un servidor backend, lo que te permite hacer peticiones a una API REST sin necesidad de crear un servidor real.

- Instala JSON Server de manera global con el siguiente comando:

```
sudo npm install -g json-server
```

- Crea un archivo llamado `data.json` en la carpeta "public". Un ejemplo de `data.json` podría ser:

```
{
  "aficiones": [
    { "id": 1, "title": "Primer post", "content": "Contenido del primer post" },
    { "id": 2, "title": "Segundo post", "content": "Contenido del segundo post" }
  ]
}
```

- Ahora, puedes iniciar JSON Server con el siguiente comando para que simule un backend:

```
npx json-server public/data.json
```

Esto hará que el servidor corra en `http://localhost:3000`, y podrás hacer peticiones a rutas como `http://localhost:3000/aficiones` para obtener los datos.

## 4. Estructura de un Proyecto React

```
proyecto-react-vite/  
├─ node_modules/  
├─ public/  
│   ├─ img/  
│   │   ├─ imagen1.jpg  
│   │   ├─ imagen2.png  
│   │   └─ ... (otras imágenes)  
│   ├─ data.json  
│   └─ favicon.ico  
├─ src/  
│   ├─ components/  
│   │   ├─ ExampleComponent.jsx  
│   │   └─ ... (otros componentes)  
│   ├─ styles/  
│   │   ├─ app.css  
│   │   ├─ ExampleComponent.css  
│   │   └─ ... (otros estilos)  
│   ├─ App.jsx  
│   ├─ main.jsx  
│   └─ index.html  
├─ .gitignore  
├─ eslint.config.js  
├─ package.json  
└─ package-lock.json
```

```
|— README.md
|— vite.config.js
```

src/index.html

El archivo HTML principal, se añade el main.jsx en "script":

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>React con Vite</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

src/main.jsx

El punto de entrada principal de tu aplicación:

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import './styles/app.css';

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

src/App.jsx

El componente principal de la aplicación:

```
import React from 'react';
import './styles/app.css';

const App = () => {
  return (
    <div className="App">
      <h1>Hola, React con Vite</h1>
    </div>
  );
};

export default App;
```

src/styles/app.css

Un ejemplo básico de estilos:

```
body {
  margin: 0;
  font-family: Arial, sans-serif;
  background-color: #f5f5f5;
  color: #333;
}

h1 {
  text-align: center;
  margin-top: 50px;
}
```

public/data.json

Un archivo JSON que puedes usar con Axios y JSON Server:

```
{
  "items": [
```

```
    { "id": 1, "name": "Item 1", "description": "Descripción  
del Item 1" },  
    { "id": 2, "name": "Item 2", "description": "Descripción  
del Item 2" },  
    { "id": 3, "name": "Item 3", "description": "Descripción  
del Item 3" }  
  ]  
}
```