



REACTJS

¿QUÉ ES REACT?



JS

React es una librería Javascript focalizada en el desarrollo de interfaces de usuario. Así se define la propia librería y evidentemente, esa es su principal área de trabajo. Sin embargo, lo cierto es que en React encontramos un excelente aliado para hacer todo tipo de aplicaciones web, SPA (Single Page Application) o incluso aplicaciones para móviles.



¿POR QUE REACT?

Alrededor de React existe un completo ecosistema de módulos, herramientas y componentes capaces de ayudar al desarrollador a cubrir objetivos avanzados con relativamente poco esfuerzo.



facebook®



UN POCO DE HISTORIA

React es una librería desarrollada inicialmente por Facebook. Es software libre y a partir de su liberación acapara una creciente comunidad de desarrolladores y entusiastas. Su creación se realizó en base a unas necesidades concretas, derivadas del desarrollo de la web de la popular red social. Además de facilitar el desarrollo ágil de componentes de interfaces de usuario, el requisito principal con el que nació React era ofrecer un elevado rendimiento, mayor que otras alternativas existentes en el mercado.

OBJETIVO DE REACT

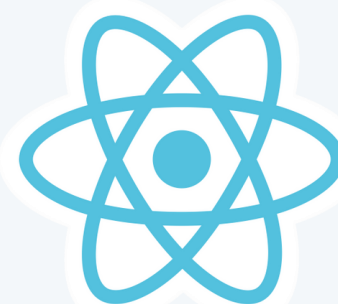
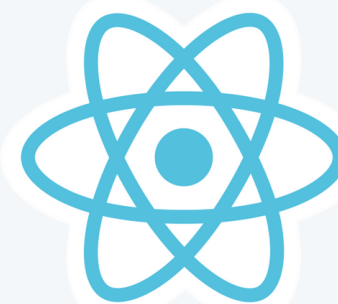
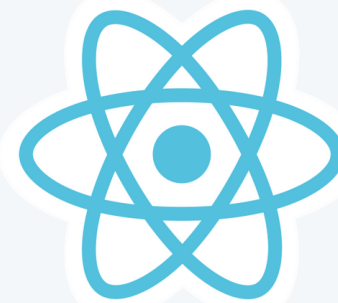
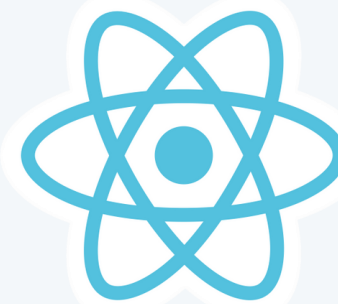
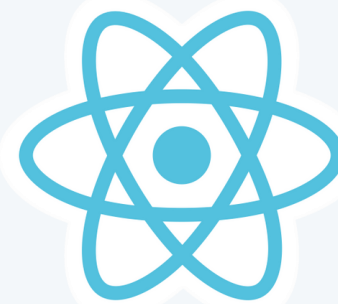
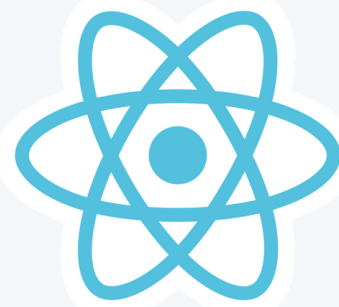


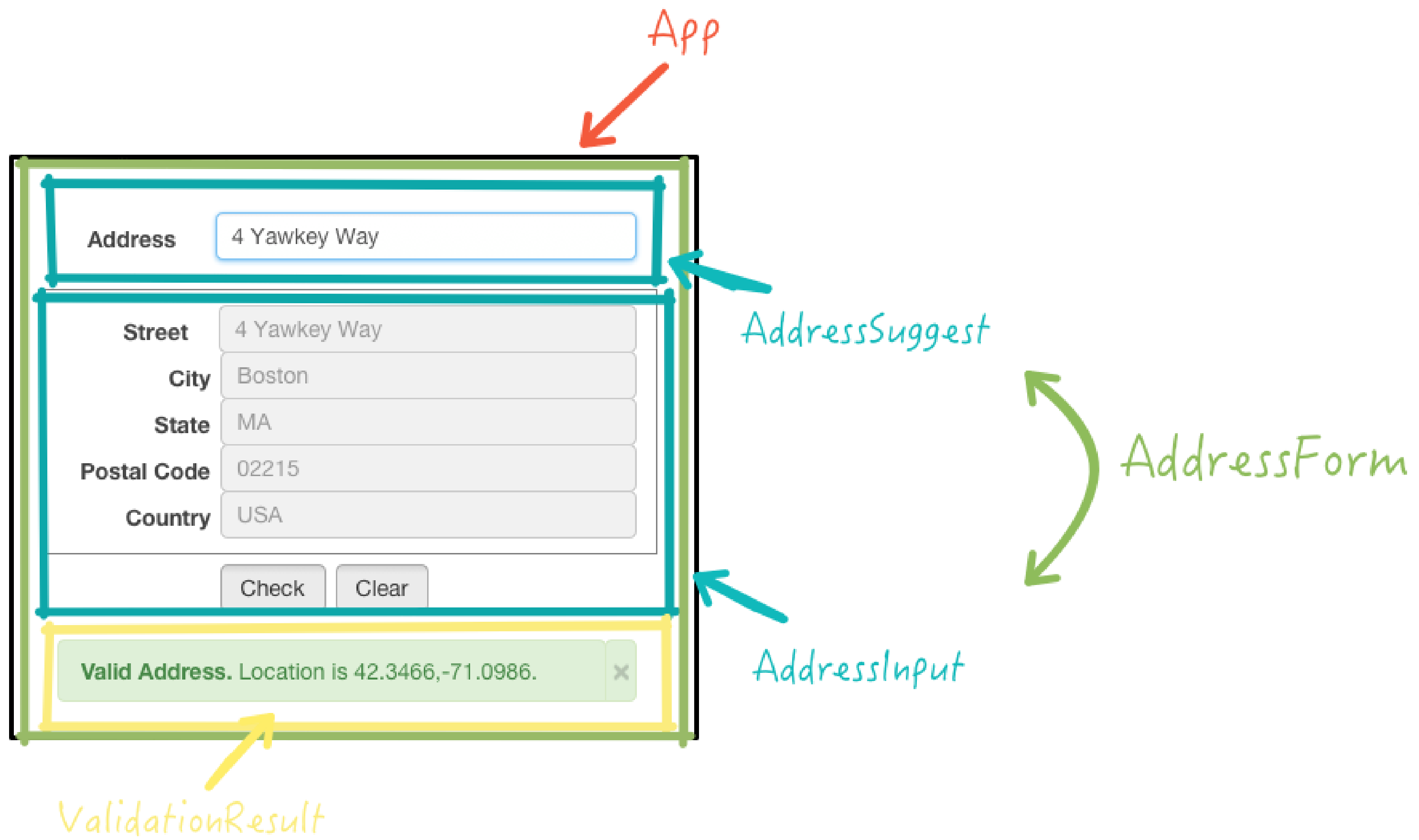
Sirve para desarrollar aplicaciones web de una manera más ordenada y con menos código que si usas Javascript puro o librerías como jQuery centradas en la manipulación del DOM. Permite que las vistas se asocien con los datos, de modo que si cambian los datos, también cambian las vistas.

¡HOLA MUNDO!

¡HOLA COMPONENTES!

El código que se suele producir mediante librerías como jQuery se pretende arquitecturizar y el modo de conseguirlo es a través de componentes. Una interfaz de usuario es básicamente creada a partir de un componente, el cual encapsula el funcionamiento y la presentación. Unos componentes se basan además en otros para solucionar necesidades más complejas en aplicaciones. También permite crear otras piezas de aplicación cómodamente, como los test.



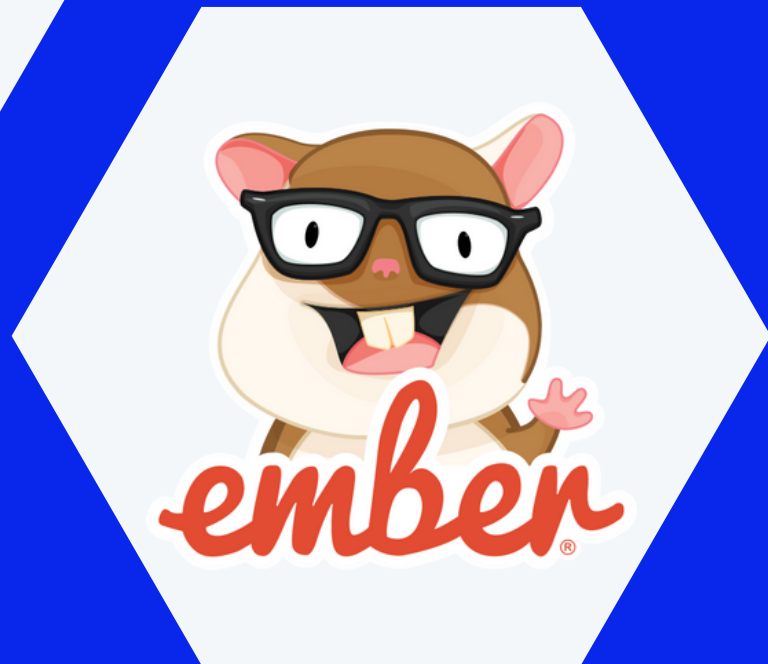


EVOLUCIÓN DE JQUERY

ReactJS solapa por completo las funcionalidades de jQuery, por lo que resulta una evolución natural para todos los sitios que usan esa librería. Podrían convivir pero no es algo que realmente sea necesario y recargaría un poco la página, por lo que tampoco sería muy recomendable.



¿Comparado con algún Framework?



En comparación con frameworks como es el caso de Angular o Ember, React se queda a mitad de camino, pues no incluye todo lo que suele ofrecer un framework completo. Pero ojo, a partir de todo el ecosistema de React se llega más o menos a las mismas funcionalidades, así que es una alternativa perfecta.

¿QUE ES?

En líneas generales podemos decir que el virtual DOM es una representación del DOM pero en memoria, que usa React para aumentar sensiblemente el rendimiento de los componentes y aplicaciones front-end.

VIRTUAL DOM



¿CÓMO FUNCIONA?

Básicamente hace que, cuando se actualiza una vista, React se encargue de actualizar el DOM Virtual, que es mucho más rápido que actualizar el DOM del navegador (DOM real). Cuando React compara el DOM Virtual con el DOM del navegador sabe perfectamente qué partes de la página debe actualizar y se ahorra la necesidad de actualizar la vista entera.

VIRTUAL DOM



ISOMORFISMO

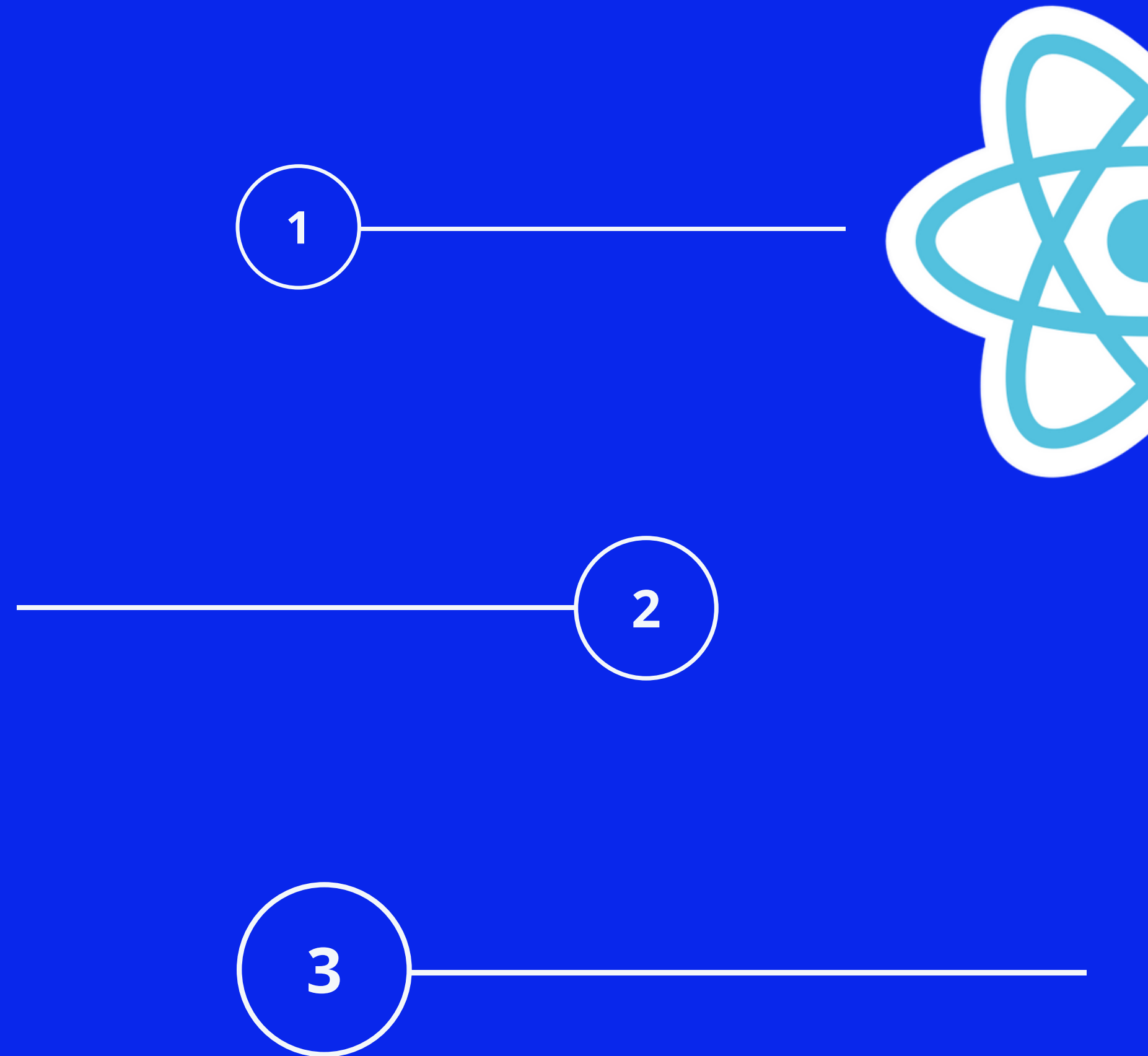
¿POR QUE REACT ES ISOMÓRFICO?

Éste es un concepto relativamente nuevo, pero muy interesante en el desarrollo de aplicaciones que se desean tengan un buen posicionamiento en buscadores. Básicamente se trata de, con un mismo código, renderizar HTML tanto en el servidor como en el cliente, rebajando la carga de trabajo necesaria para realizar aplicaciones web amigables para buscadores.

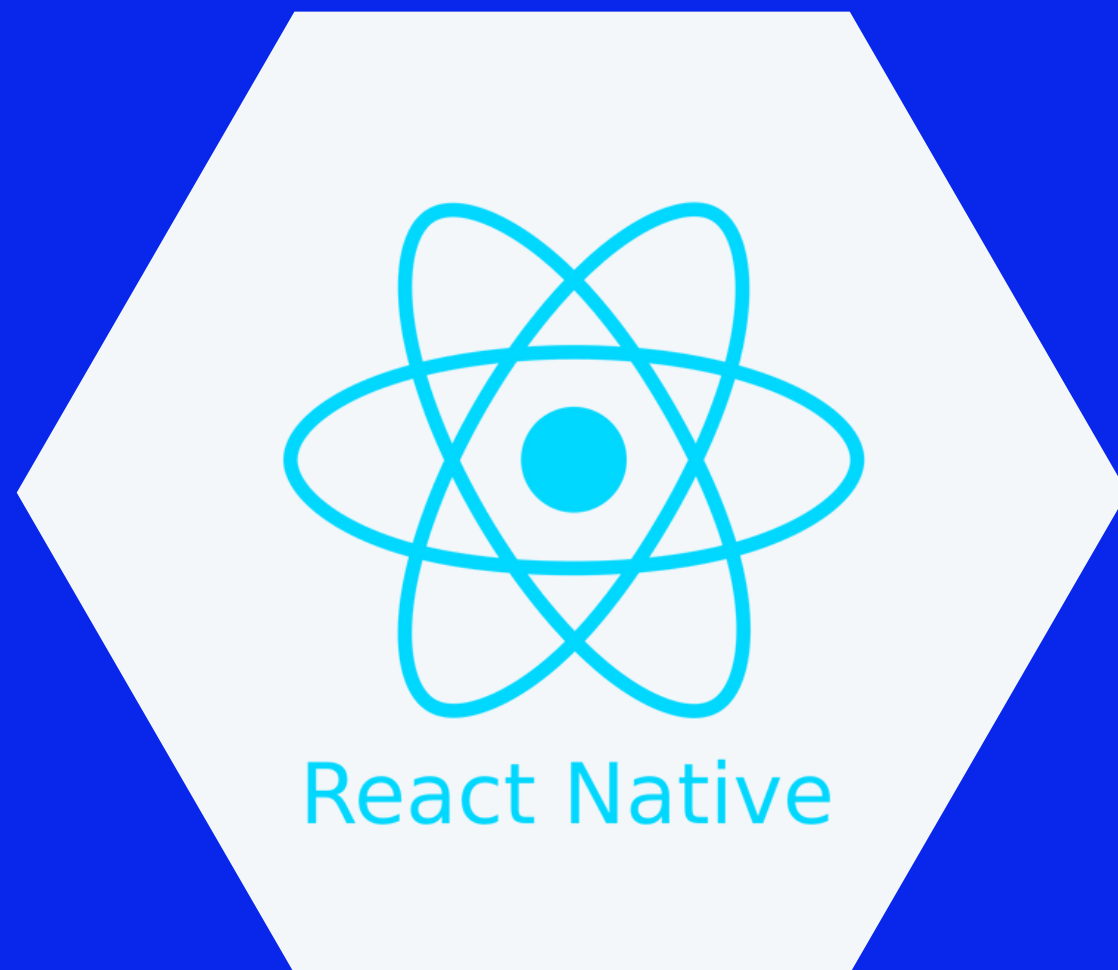


ECOSISTEMA DE REACT

- 1 REDUX
- 2 NODE_MODULES
- 3 ROUTING



REACT NATIVE



React Native es otra de las herramientas disponibles en el ecosistema, que permite llevar una aplicación escrita con Javascript y React como aplicación nativa para dispositivos iOS, Android, etc. Y se trata de aplicaciones nativas! el código Javascript con React se compila a nativo, en lugar de usar web views como ocurre generalmente en el desarrollo híbrido.

REACT TYPESCRIPT

Si bien TypeScript es el lenguaje de Angular, esto no significa que no se pueda usar en conjunto con el competidor directo de Angular, es decir React.js. Entre las muchas ventajas de usar TypeScript con React se encuentra el tener una aplicación más robusta y a prueba de bugs, reemplazar el uso de PropTypes, etc

INSTALACIÓN



1 . INSTALAR NODEJS

<https://nodejs.org/es/>

2 . INSTALAR CREATE-REACT-APP

```
npm install -g create-react-app
```

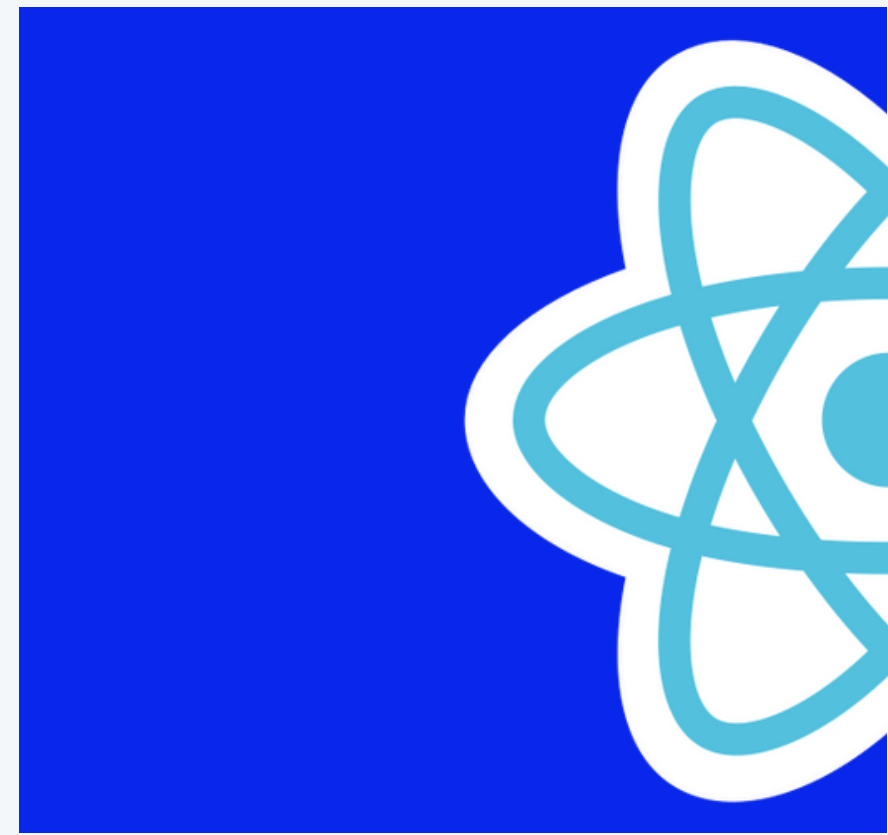
3 . CREAR PROYECTO

```
create-react-app [nombre_proyecto]
```

4 . INICIAR PROYECTO

```
npm start (Sobre la carpeta del proyecto)
```

ESTRUCTURA DEL PROYECTO



```
README.md
node_modules/ // Librerías de nuestra aplicación
package.json // Información y dependencias de nuestro proyecto
.gitignore // Fichero donde se determina qué carpetas no deberán ser trackeadas con git
public/
favicon.ico
index.html // Página donde se inyectarán los componentes de React
src/ // Directorio de desarrollo que es el que usaremos nosotros
  App.css
  App.js
  App.test.js
  index.css
  index.js
  logo.svg
```

INGRESAR A HTTP://LOCALHOST:3000



Edit `src/App.js` and save to reload.

[Learn React](#)

App.js

```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="App">
        <div className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h2>Welcome to React</h2>
        </div>
        <div className="App-intro">
        </div>
      </div>
    );
  }
}

export default App;
```

