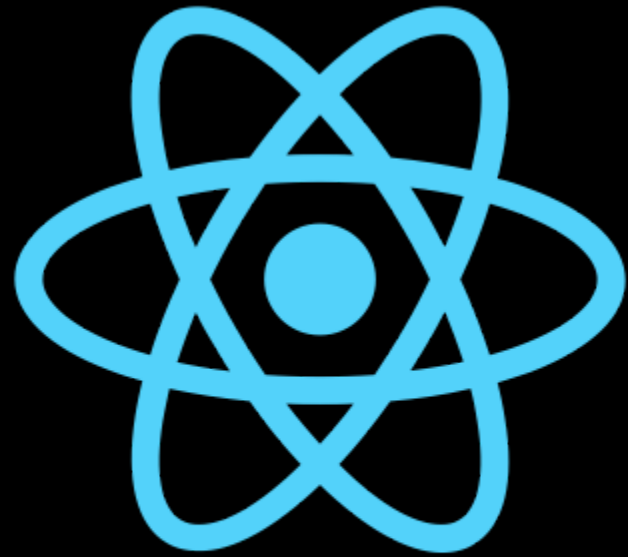


HENRY

A bright yellow beam of light originates from the left edge of the frame and points towards the letter 'R' in the word 'HENRY'. The beam is wider on the left and tapers as it moves to the right. The word 'HENRY' is rendered in a bold, black, sans-serif typeface.



React

Parte I



REACT



React es una librería de **JavaScript** que es **declarativa**, **eficiente** y **flexible** y sirve para construir interfaces de usuarios. Esta librería fue creada por el equipo de *facebook* e *instagram*, que fue liberada y ahora es un proyecto *open source*.



REACT

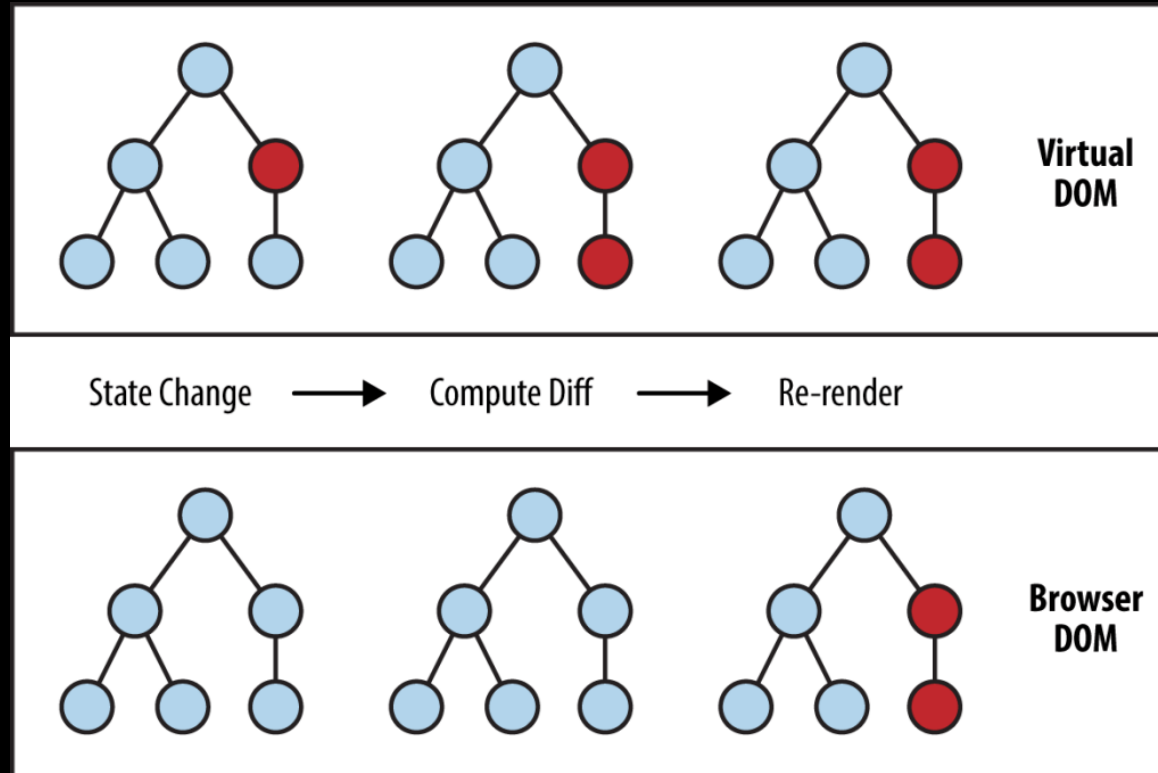
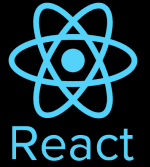


```
1
2 const numbers = [4,2,3,6];
3
4 //imperativo (le decimos COMO queremos que se hagan las cosas)
5
6 let total = 0;
7 for (let i = 0; i < numbers.length; i++){
8   total += numbers[i]
9 }
10
11
12
13 //declarativo (decime QUE queremos que se haga)
14
15 numbers.reduce(function(p, c){
16   return p + c;
17 })
```

Programación declarativa vs imperativa



REACT



Virtual DOM



REACT



TfL Tube Tracker

`<Network />`

Bakerloo Line
`<Line />`
Stations Go

Central Line
`<Line />`
Stations Go

Circle Line
`<Line />`
Stations Go

District Line
`<Line />`
Stations Go

Hammersmith & City Line
`<Line />`
Stations Go

Jubilee Line
`<Line />`
Stations Go

Metropolitan Line
`<Line />`
Stations Go

Station Name `<Predictions />`

Platform 1 `<DepartureBoard />`

Destination	Due	Current location
White City	0:00	At Platform
Ealing Broadway	2:00	Holland Park
West Ruislip	4:30	Notting Hill Gate
Ealing Broadway	6:00	Queensway

`<Trains />`

Platform 2 `<DepartureBoard />`

Destination	Due	Current location
White City	0:00	At Platform
Ealing Broadway	2:00	Holland Park
West Ruislip	4:30	Notting Hill Gate
Ealing Broadway	6:00	Queensway

Component Driven Development

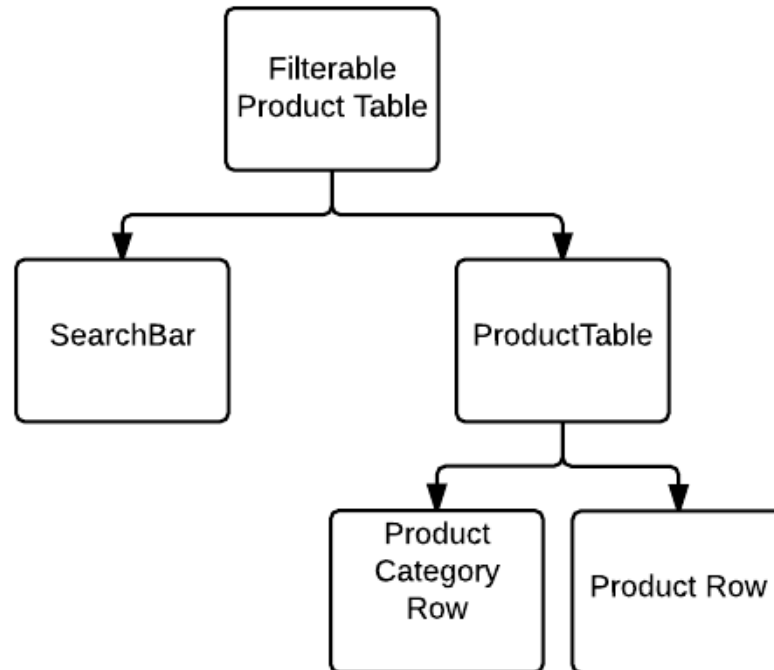


REACT



☐ Only show products in stock

Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99



single responsability principle, o princio de responsabilidad única



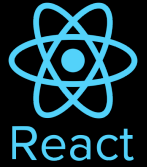
Componente



state is used for internal communication inside a Component



JSX



```
1
2 const element = <h1>Hello, world!</h1>;
```



Babel compila JSX a llamadas de
React.createElement().



```
1 "use strict";
2
3 var element = /*#__PURE__*/React.createElement("h1", null, "Hello, world!");
```



JSX



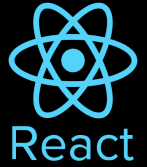
```
1 function formatName(user) {  
2   return user.firstName + ' ' + user.lastName;  
3 }  
4 const user = {  
5   firstName: 'Toni',  
6   lastName: 'Tralice'  
7 };  
8 const element = (  
9   <h1>  
10     Hello, {formatName(user)}!  
11   </h1>  
12 );
```

BABEL

```
1 "use strict";  
2  
3 function formatName(user) {  
4   return user.firstName + ' ' + user.lastName;  
5 }  
6  
7 var user = {  
8   firstName: 'Toni',  
9   lastName: 'Tralice'  
10 };  
11 var element = React.createElement("h1", null, "Hello, ", formatName(user), "!");
```



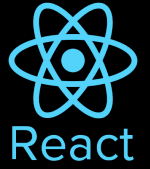
Componentes Funcionales y De Clase



```
1 // Funcional
2
3 function Welcome(props) {
4   return <h1>Hello, {props.name}</h1>;
5 }
```



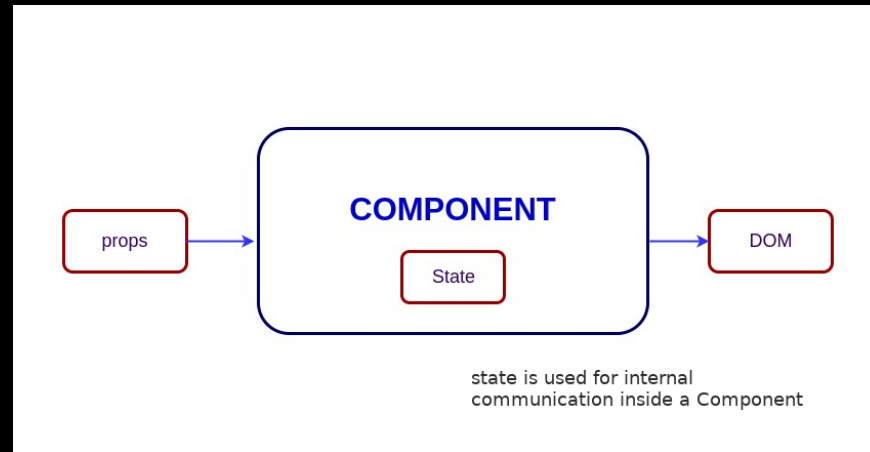
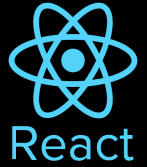
```
1 // Componente de Clase
2
3 class Welcome extends React.Component {
4   render() {
5     return <h1>Hello, {this.props.name}</h1>;
6   }
7 }
```



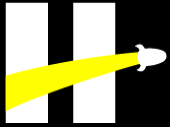
<Demo />



Props



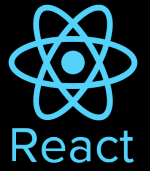
Conceptualmente, los componentes son como funciones de JS. Aceptan inputs arbitrarios (props) y retornan elementos REACT que representan lo que debería aparecer en la pantalla



Props



```
1
2
3 class Welcome extends React.Component {
4   render() {
5     return <h1>Hello, {this.props.name}</h1>;
6   }
7 }
8
9 function Welcome(props) {
10   return <h1>Hello, {props.name}</h1>;
11 }
```



<Demo />



JSX a JS



jsx

```
<h1 className="greeting">  
  Hello, world!  
</h1>
```

js

```
React.createElement(  
  'h1',  
  {className: 'greeting'},  
  'Hello, world!'  
);
```

@babel/preset-react



Webpack

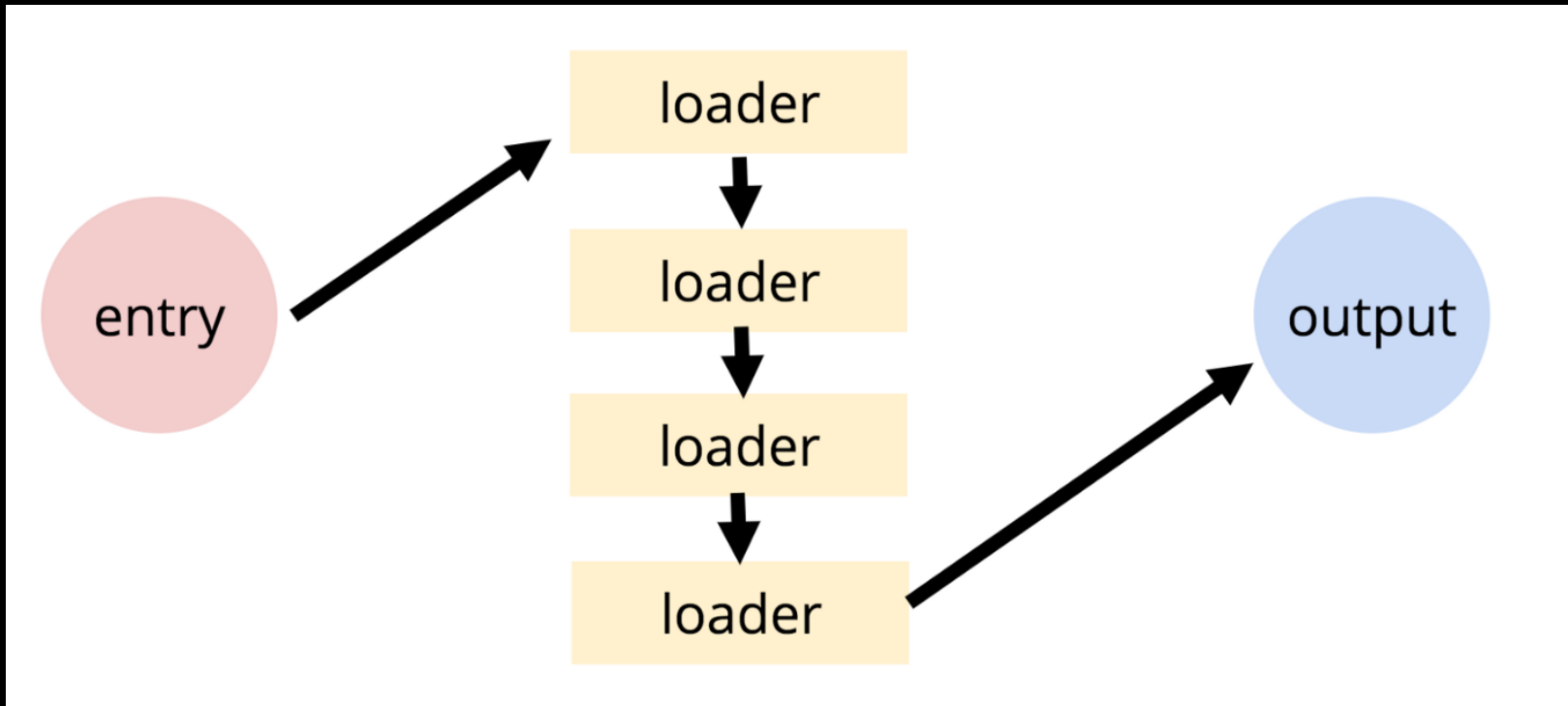
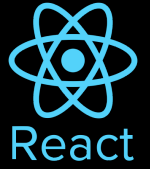


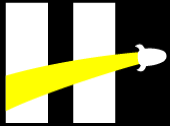
```
1 // instalamos webpack
2 npm i -D webpack webpack-cli
3
4 // instalamos React
5
6 npm i react react-dom
7
8 // instalamos babel y sus dependencias
9
10 npm i -D @babel/core @babel/preset-env @babel/preset-react babel-loader
```

- **webpack:** La base de webpack, el bundler y el superhéroe de la historia.
- **webpack-cli:** Para habilitar comandos de webpack en la consola.
- **react:** La librería de react.
- **react-dom:** La otra librería de react para trabajar con el DOM.
- **@babel/core:** La base de babel, el compilador de javascript moderno.
- **@babel/preset-env:** Para que babel pueda compilar ECMAScript 6.
- **@babel/preset-react:** Para que babel pueda compilar jsx.
- **babel-loader:** Para que webpack utilice babel.

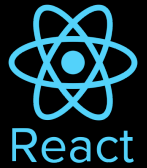


Webpack

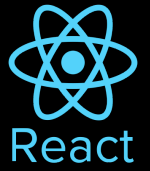




Webpack



```
1 // webpack.config.js
2
3 module.exports = {
4   entry: 'index.js',
5   output: {
6     path: __dirname + '/dist',
7     filename: '[name].js'
8   },
9   module: {
10    rules: [
11      {
12        test: /\.js$/,
13        exclude: /node_modules/,
14        use: {
15          loader: 'babel-loader',
16          options: {
17            presets: ['@babel/preset-react', '@babel/preset-env']
18          }
19        }
20      }
21    ]
22  }
23 }
```



<Demo />