



Vue.js

# # Características

- **Framework Progresivo**

# # Características

- **Framework Progresivo**
- **MV\***

# # Características

- **Framework Progresivo**
- **MV\***
- **Reactivo**

# # Características

- **Framework Progresivo**
- **MV\***
- **Reactivo**
- **Vue.js core → UI**

# # Características

- **Framework Progresivo**
- **MV\***
- **Reactivo**
- **Vue.js core → UI**
- **Componentes**

# # Características

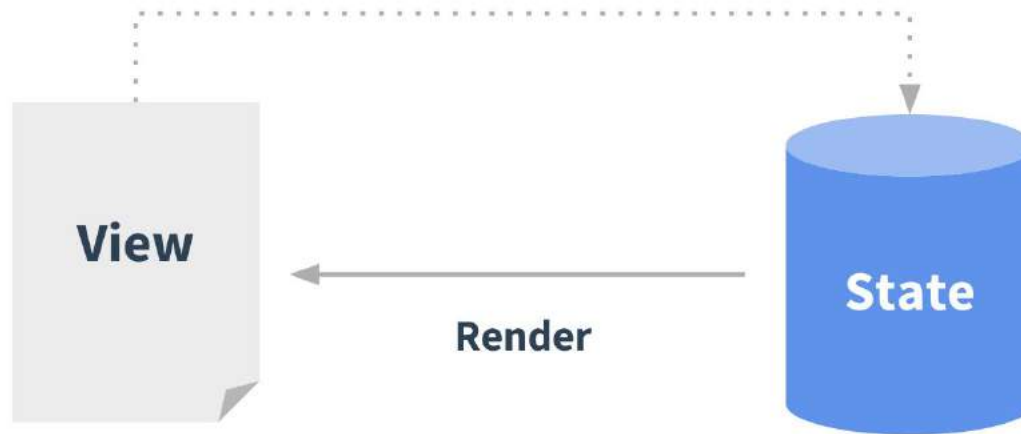
- **Framework Progresivo**
- **MV\***
- **Reactivo**
- **Vue.js core → UI**
- **Componentes**
- **Fácil de aprender**

# # Características

- **Framework Progresivo**
- **MV\***
- **Reactivo**
- **Vue.js core → UI**
- **Componentes**
- **Fácil de aprender**
- **Vue.js + herramientas + librerías = SPA**

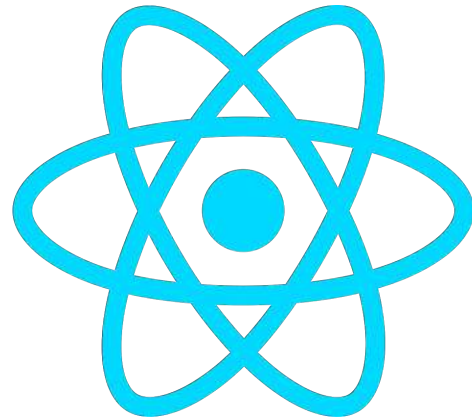


**User Input**

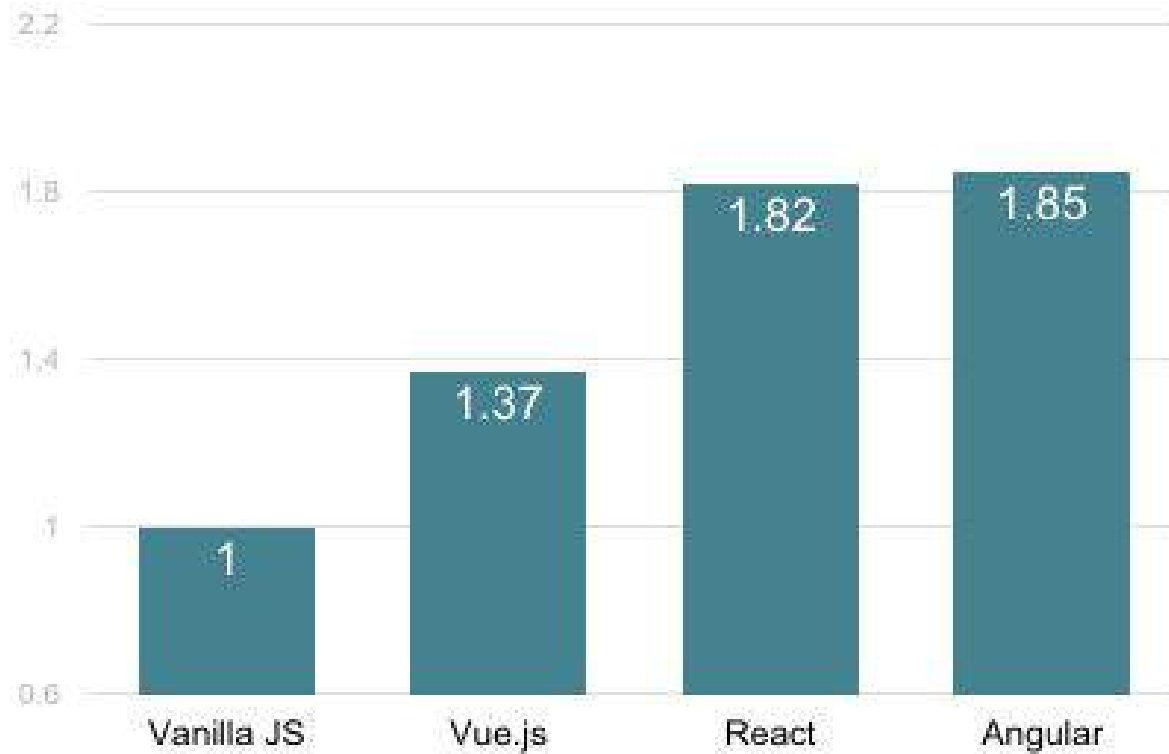




vs



## Performance



<http://stefankrause.net/js-frameworks-benchmark4/webdriver-ts/table.html>

# # Vue.js vs React

## Similitudes

- Virtual DOM
- Componentes
- Librería core sólo para la UI

## Performance

- Minimizar las interacciones con el DOM
- Minimizar los cálculos de JavaScript para saber qué modificar

## Update performance

- `shouldComponentUpdate` y estructuras de datos inmutables
- Dependencias auto-trackeadas

## Developer Experience

# # Vue.js vs React

## HTML

- React = JavaScript everywhere (JSX)

```
render () {  
  let { items } = this.props  
  
  let children  
  if (items.length > 0) {  
    children = (  
      <ul>  
        {items.map(item =>  
          <li key={item.id}>{item.name}</li>  
        )}  
      </ul>  
    )  
  } else {  
    children = <p>No items found.</p>  
  }  
  
  return (  
    <div className='list-container'>  
      {children}  
    </div>  
  )  
}
```

# # Vue.js vs React

## HTML

- Vue.js = templates

```
<template>
  <div class="list-container">
    <ul v-if="items.length">
      <li v-for="item in items">
        {{ item.name }}
      </li>
    </ul>
    <p v-else>No items found.</p>
  </div>
</template>
```

# # Vue.js vs React

## CSS-in-JS

```
1. const styles = {  
2.   'btn': {  
3.     'display': 'inline-block',  
4.     'border': '0',  
5.     'padding': '6px 12px'  
6.   },  
7.  
8.   'btn_primary': {  
9.     'color': '#fff',  
10.    'backgroundColor': '#f74A27'  
11.  },  
12.  'btn_primary_hover': {  
13.    'backgroundColor': '#ff7857'
```

# # Vue.js vs React

## Component-Scoped CSS

```
<style scoped>
  @media (min-width: 250px) {
    .list-container:hover {
      background: orange;
    }
  }
</style>
```



# # Vue.js vs React

## Component-Scoped CSS

```
<style scoped>
  @media (min-width: 250px) {
    .list-container:hover {
      background: orange;
    }
  }
</style>
```

`.list-container:hover` → `.list-container[data-v-21e5b78]:hover`

# # Vue.js vs React

**Redux ↔ Vuex**

**React Router ↔ vue-router**

# # Vue.js vs Angular

- **TypeScript**
- **Performance y tamaño**
- **Flexibilidad**
- **Curva de aprendizaje**