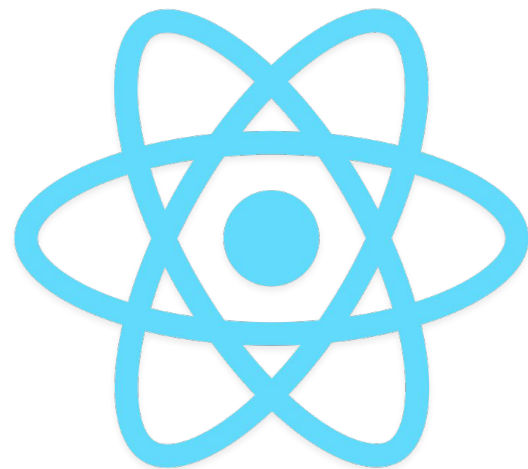




VS



Vue.js

React.js

Introducción

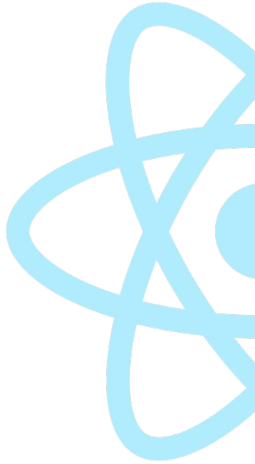
¿Qué son?

¿Librería o Framework?

Evolución

Comparativas

Instalación



¿Qué son?

Son **framework** de **JavaScript reactivos**, ésto quiere decir que la **comunicación** entre el **modelo** y la **vista** es en “**tiempo real**”.

Ambos usan la **programación** por **componentes**, lo cual nos permite **dividir** nuestra **web** en **pequeños módulos**.



¿Librería o Framework?

Librería: proporcionan funciones/métodos muy simples que implementamos en nuestro código.

Framework: dispone de una estructura completa y con un patrón de diseño al que tendremos que adaptar nuestro código.

Evolución

2.610

63.544

React.js

2013

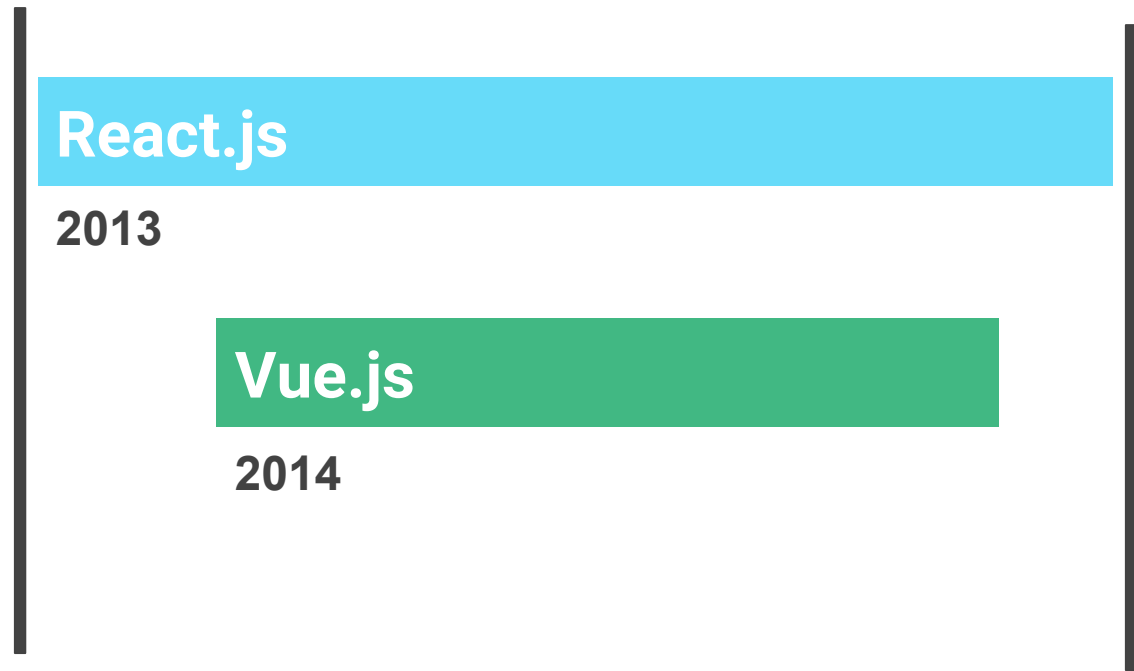
Vue.js

2014



Git-Hub Proyectos

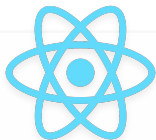
2017



Comparativas



Similitudes



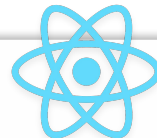
- Son framework reactivos.
- Arquitectura basada en componentes.
- Uso DOM virtual.
- Agilizan el proceso de desarrollo.
- Ayudan a realizar proyectos más seguros, ya que presentan actualizaciones constantes.



Ventajas Vue js

- No necesita la creación de un proyecto.
- + Facilidad a la hora de aprenderlo.

Ventajas React js



- React Native, para la creación de apps nativas para móviles.
- Mejor performance.



Instalación




Instalación

NPM



```
$ npm install vue
```

```
$ npm install -g create-react-app  
$ create-react-app mywebapp  
$ cd mywebapp  
$ npm start
```




CDN



```
<script  
src="https://cdn.jsdelivr.net/npm/  
vue@2.5.13/dist/vue.js">  
</script>
```

```
<script crossorigin  
src="https://unpkg.com/react@16  
/umd/react.production.min.js"></s  
cript>
```





Vue.js

Iniciación a Vue

Instancia Vue

Directivas

Eventos

Bindings

Componentes

Vue Routing

Axios

Demo final





Instancia Vue

Instancia Vue

```
new Vue({  
  el: '#app',      //elemento html, donde actuará la instancia de vue  
  data:{  
    //Declaración de variables  
  },  
  methods:{  
    //Declaración de funciones/métodos  
  },  
  computed:{  
    //Declaración de propiedades computadas  
  }  
});
```



Directivas

Directivas Vue

Se utilizan para indicarle a vue que tiene que hacer acciones especiales en el DOM, su sintaxis siempre empieza de la misma forma **v-** , seguido de la acción a realizar.

v-show //muestra u oculta elementos html, pero no afecta al DOM

v-if , **v-else-if** , **v-else** //muestra o elimina elementos html en el DOM

v-bind //bindea los atributos de las etiquetas html para añadir un valor

v-on //usa manejadores de eventos

v-for //recorre un array, lista ,contador ...

v-model // Establece una conexión bidireccional entre el modelo y la vista ... → Is Magic

DEMO

-Directivas-



Eventos

Eventos Vue

Modificadores

.stop //stop propagación
.prevent //event default
.capture //stop propagación
...

v-on:click

v-on:keyup

v-on:focus

...

.enter
.tab
.delete
...

DEMO

-Eventos-



Bindings

Bindings Vue

Se usa para incrustar un valor de una variable en el atributo de un elemento del DOM.

v-bind:class=" "

v-bind:style=" "

v-bind:type=" " //No admite tipos dinámicos.

...

DEMO

-Bindings-



Componentes

Componentes Vue

Es una extensión de la instancia de vue, que nos permitirá agrupar cierto código, para su reutilización.

Sintaxis creación componente vue en JS

```
Vue.component('mi-componente',  
{  
  // opciones  
})
```

Sintaxis uso componente vue en HTML

`<mi-componente>`

`</mi-componente>`

DEMO

-Componentes-



ROUTING

Vue Routing

Es un paquete que se integra con Vue, para facilitar la creación de webs SPA(Single page application)


NPM

```
$ npm install vue-router
```



CDN

```
<script  
src="/ruta/a/vue-rou  
ter.js"></script>
```



Vue Routing

Creación instancia Vue Router

```
1 const routes = [  
  { path: '/home', component: home },  
  { path: '/about', component: about }  
];  
  
2 const router = new VueRouter({  
  routes // array de object router  
});  
  
3 new Vue({  
  el: '#app'  
  router: router // añadimos a la instancia principal, la instancia VueRouter  
});
```

DEMO

-Vue Routing



AXIOS

Axios Vue

Es una librería que facilita la conexión asíncrona de ajax con un servidor web.

NPM

```
$ npm install axios
```



CDN

```
<script  
src="https://unpkg.co  
m/axios/dist/axios.min  
.js"></script>
```



Axios Vue

Implementación de axios

```
axios.get('/servidorRemoto')  
  .then(function (resultado) {  
    //Cuando todo sale OK  
  })  
  .catch(function (error) {  
    //Cuando ocurre un error  
  });
```

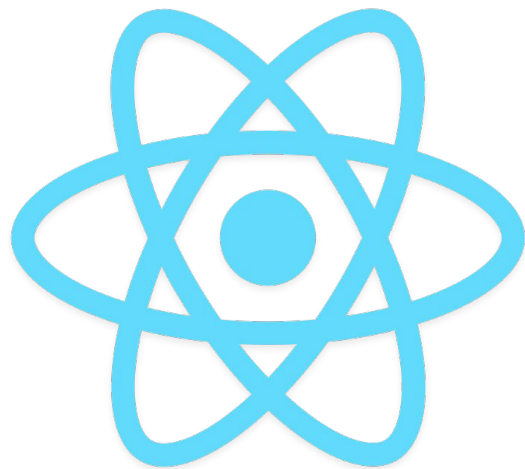
```
axios.post('/servidorRemoto',{  
  nombre : 'Pepito'  
  apellido : 'Golondrino'  
})  
  .then(function (resultado) {  
    //Cuando todo sale OK  
  })  
  .catch(function (error) {  
    //Cuando ocurre un error  
  });
```

DEMO

-Vue Axios

DEMO

FINAL VUE



React.js

Iniciación a React

DOM Virtual

Render

Props

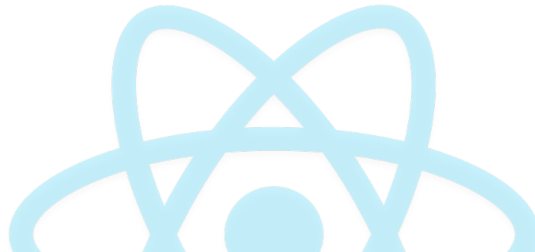
Bindings

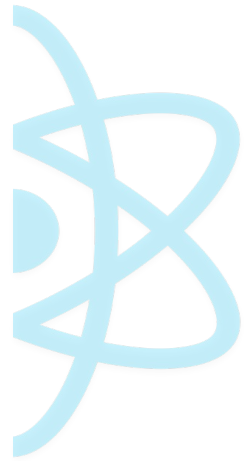
Componentes

Estados

Eventos

Demo final

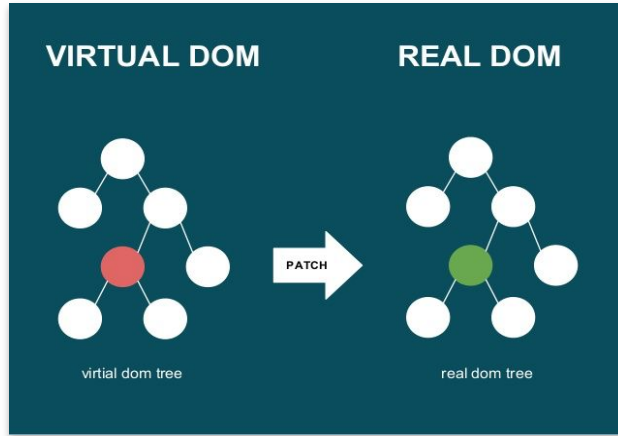


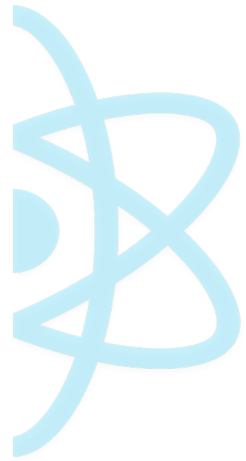


DOM Virtual

DOM Virtual

Objeto Javascript que simula el DOM , comprueba los cambios respecto al DOM original y solo repinta los nodos donde se hayan producido cambios.





Componentes

Componentes

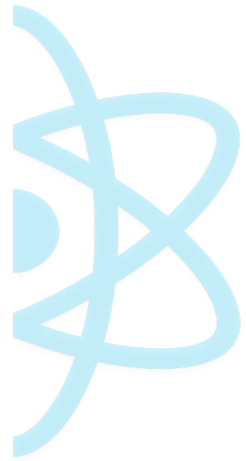
Elementos que constituyen la interfaz del usuario (un botón, un buscador..). Reaccionan a los eventos que se producen y se repintan cuando cambia su estado.

Creación componente React en JS

```
//Importa component desde React
import React, { Component } from 'react';
//Crea una clase Nota
class Nota extends Component {
  render() {
    return (/*Código*/ );
  }
}
export default Nota;
```

Uso del componente en html

<Nota/>



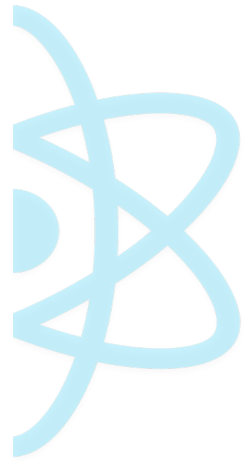
Render

Render

Construye el código HTML que devuelve `return()`

Cuando ya se ha usado anteriormente la función `render()`, al volver a usarla se realizará una actualización del DOM para reflejar los últimos cambios.

```
render() {  
  return(  
    <p>{this.props.titulo}<p>  
  );  
}
```

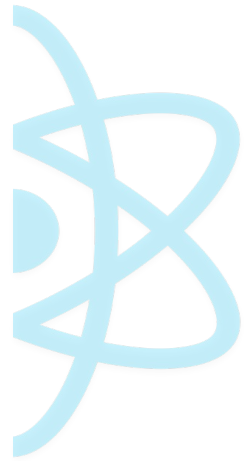


Estados

Estados

Distintas propiedades que puede tener un componente.

```
constructor(){  
  super();  
  //asigna propiedades al estado del componente  
  this.state={  
    notas:[  
      {titulo: "Preparar sysmana 2018", tarea: "Ir a la  
sysmana 2018!"},  
      {titulo: "Hacer BBDD", tarea: "Hacer BBDD"}  
    ]  
  }  
}
```

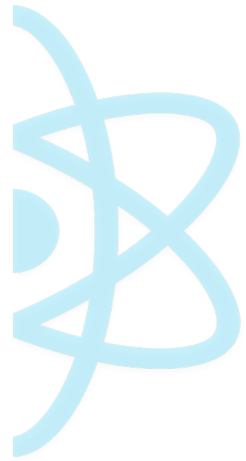


Props

Props

Son propiedades añadidas a los componentes cuando son creados. Son establecidas en el padre y están disponibles a lo largo de la vida de un componente. Se puede acceder a ellos mediante `this.props`

```
class Componente{  
  render(){  
    return(  
      //Asignamos la propiedad "titulo" al props de "Nota"  
      <button onclick={this.props.titulo}>  
    );  
  }  
}
```



Bindings

Bindings

La función `.bind()` sirve para pasar argumentos al uso de un callback.

1

```
funcion1(){  
  //funcionamiento  
}
```

2

```
//Callback de la “funcion1” con bind  
<componente  
  onAdd={this.funcion1.bind(parametro)}/>
```

DEMO
FINAL REACT

¿Dónde encontrarnos?



@susomejias



@susomejias



@jponferrada26



@jponferrada26



@miguel5gavilan



@miangame



@JuanRueMor



@Juan-ruedaa



¡GRACIAS!

¿Preguntas?

