



VUE FRAMEWORK

Breve historia de Vue

- Vue fue lanzado en 2014 convirtiéndose en el framework más joven dentro de los más utilizados.
- Fue desarrollado por una única persona, Evan You, mediante la plataforma Patreon, que permite apoyar a creadores de contenido de todo tipo.
- La idea de Evan era crear un framework que combinara “los mejores enfoques de Angular, Ember y React” con otras características que hicieran escritura de páginas web “rápida sencilla y disfrutable”.



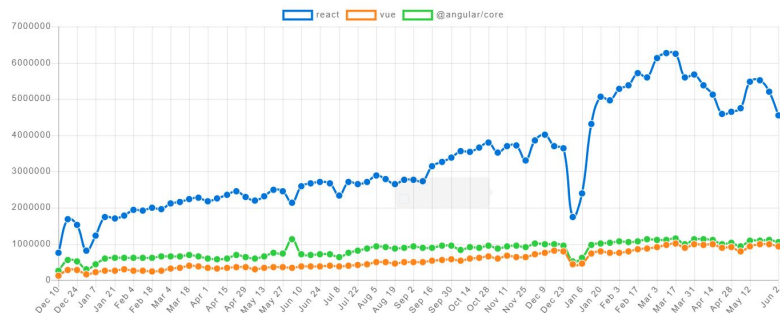
Evan You, ex empleado de Google, formaba parte de su equipo de Angular.

Algunos datos sobre Vue



2019 State of JS Report

Downloads in past 2 Years



Descargas en 2019

What JavaScript frameworks do you regularly use, if any? (%)



Frameworks usados regularmente
2019 State of Developer Ecosystem Report

Explicación conceptual básica

¿Que es vue?

Es un framework pensado para trabajar diseño UI desde el lado front haciendo la manipulación del dom sencilla y práctica.

Utiliza un patrón MVVM (model–view–viewmodel) es decir, separa la interfaz del usuario de la lógica del usuario. Vue utiliza una clase a la que va dando propiedades por lo que se abstrae de la vista.

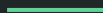
Cómo funciona

La base de Vue es hacer una instancia Vue a la que le pasamos una variable. Esa variable será el “ámbito” en el que podremos utilizar la lógica del objeto Vue.

Una vez tenemos nuestra instancia hecha, utilizaremos etiquetas para añadir datos, funciones, estilos y variables. Las etiquetas utilizadas para ello son: El, data y methods.

Elementos de la instancia Vue

- El
- data
- methods



El (Element)

Elemento con el que vamos a trabajar. Es el contenedor donde se va a ejecutar Vue.

```
<section id="app" class="section">
  <h2 class="titulo"> Ejemplo con Vue</h2>

  <table class="table">
    <tbody>
      <tr>
        <td colspan="2"> Se han agregado 0 elementos</td>
      </tr>
      <tr>
        <td>
          <input type="text" class="input" />
        </td>
        <td>
          <button type="button" class="button">Agregar</button>
        </td>
      </tr>
    </tbody>
  </table>
</section>

<script src="https://cdn.jsdelivr.net/npm/vue@2.5.16/dist/vue.js"></script>

<script>
  const app = new Vue({
    el: "#app"
  })
</script>
```

Data

Utilizando la etiqueta v-model asignamos al input un modelo. Ese modelo está definido en data

```
<table class="table">
  <tbody></tbody>
  <tfoot>
    <tr>
      <td colspan="2"> Se han agregado 0 elementos</td>
    </tr>
    <tr>
      <td>
        <input v-model="newItem" type="text" class="input" />
      </td>
      <tr>
        <button type="button" class="button">Agregar</button>
      </tr>
    </tr>
  </tfoot>
</table>

</section>

<script src="https://cdn.jsdelivr.net/npm/vue@2.5.16/dist/vue.js"></script>

<script>
  const app = new Vue({
    el: "#app",
    data: {
      newItem: '', //establecemos el modelo y lo asignamos a un elemento
      items: [] //array donde van a estar los elementos que iremos agregando
    }
  })
</script>
```

Methods

Definimos la función add, la cual se activará cuando hagamos click en el botón. La función add está definida dentro de methods

```
<tr>
  <button @click="add" type="button" class="button">Agregar</button>
</tr>
</tfoot>
</table>

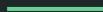
</section>

<script src="https://cdn.jsdelivr.net/npm/vue@2.5.16/dist/vue.js"></script>

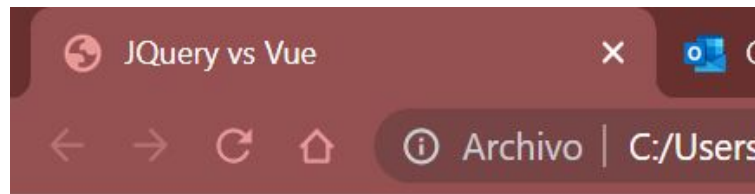
<script>
  const app = new Vue({
    el: "#app",
    data: {
      newItem: '', //establecemos el modelo y lo asignamos a un elemento
      items:[] //array donde van a estar los elementos que iremos agregando
    },
    methods:{
      add(){ //creamos la funcion add
        if(this.newItem.trim().length <= 3){
          return;
        }
        this.items.push(this.newItem);
        this.newItem = ''; //limpiamos el item para la siguiente iteracion
      }
    }
  })
})
```


Comparación Vue vs JQuery

- Código JQ
- Código Vue



Resultado de la página con JQ

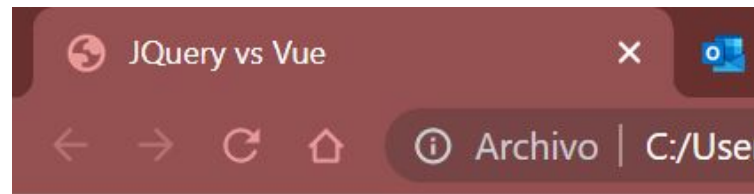


JQuery vs Vue

Ejemplo JQuery

Se han agregado 0 elementos

Agregar



JQuery vs Vue

Ejemplo JQuery

Estudiar Web

Se han agregado 1 elementos

Agregar

Código en JQuery

```
<section class="section">
  <h1 class="title">JQuery vs Vue</h1>
  <h2 class="subtitle">Ejemplo JQuery</h2>

  <table id="lista" class="table">
    <tbody></tbody>
    <tfoot>
      <tr>
        <td colspan="2">Se han agregado <span id="contador">0</span>
        elementos</td>
      </tr>
      <tr>
        <td>
          <input type="text" class="input" id="producto"/>
        </td>
        <td>
          <button id="add" type="button" class="button">Agregar</button>
        </td>
      </tr>
    </tfoot>
  </table>
</section>
```

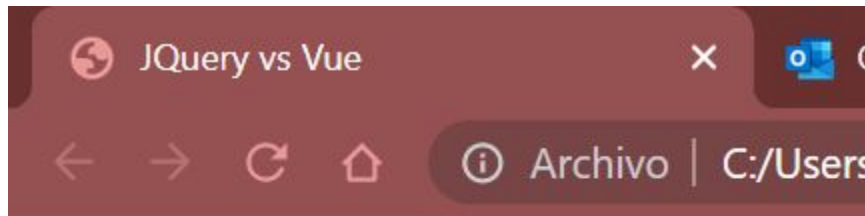
```
<script>
  $(document).ready(()=> {
    let tbody = $("#lista tbody"),
    input = $("#producto"),
    add = $("#add"),
    total = $("#contador");

    add.click(() =>{
      let tr = `
        <tr>
          <td>${input.val()}</td>
        </tr>
      `;

      if(input.val().trim().length <= 3){
        return;
      }

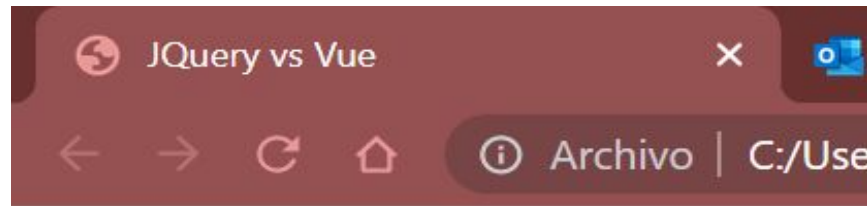
      tbody.append(tr);
      input.val('');
      total.text(tbody.find('tr').length);
    })
  })
</script>
```

Resultado de la página con Vue



Ejemplo con Vue

Se han agregado 0 elementos



Ejemplo con Vue

Estudiar Web en Vue

Se han agregado 1 elementos

Código en Vue

```
<section id="app" class="section">
  <h2 class="titulo"> Ejemplo con Vue</h2>

  <table class="table">
    <tbody>
      <tr v-for="item in items">
        <td>{{item}}</td>
      </tr>
    </tbody>
    <tfoot>
      <tr>
        <td colspan="2"> Se han agregado {{items.length}} elementos</td>
      </tr>
      <tr>
        <td>
          <input v-model="newItem" type="text" class="input" />
        </td>
        <td>
          <button @click="add" type="button" class="button">Agregar</button>
        </td>
      </tr>
    </tfoot>
  </table>
</section>
```

```
<script>
  const app = new Vue({
    el: "#app",
    data: {
      newItem: '', //establecemos el modelo y lo a
      items: [] //array donde van a estar los eleme
    },
    methods: {
      add(){ //creamos la funcion add
        if(this.newItem.trim().length <= 3){
          return;
        }
        this.items.push(this.newItem);
        this.newItem = ''; //limpiamos el item p
      }
    }
  })
</script>
```

Diferencias Vue vs JQuery

```
<script>
  $(document).ready(()=> {
    let tbody = $("#lista tbody"),
    input = $("#producto"),
    add = $("#add"),
    total = $("#contador");

    add.click(() =>{
      let tr = `
        <tr>
          <td>${input.val()}</td>
        </tr>
      `;

      if(input.val().trim().length <= 3){
        return;
      }

      tbody.append(tr);
      input.val('');
      total.text(tbody.find('tr').length);
    })
  })
</script>
```

Script en JQ

```
<script>
  const app = new Vue({
    el: "#app",
    data: {
      newItem: '', //establecemos el modelo y lo asignamos a un elemento
      items:[] //array donde van a estar los elementos que iremos agregando
    },
    methods:{
      add(){ //creamos la funcion add
        if(this.newItem.trim().length <= 3){
          return;
        }
        this.items.push(this.newItem);
        this.newItem = ''; //limpiamos el item para la siguiente iteracion
      }
    }
  })
</script>
```

Script en Vue