

# PRÁCTICA CON VUEJS

Alejandro Mos Vilar

# Índice

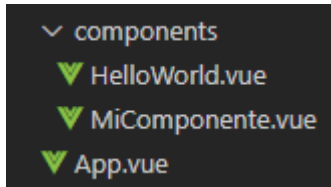
1. Práctica con VueJS.....	3
1.1. Agregar un componente nuevo a la base que nos crea vue cli de forma automática.....	3
1.2. Sobre este componente agregar un console.log ("nombre de la etapa") en cada una de las etapas del ciclo de vida de la instancia Vue. Donde "nombre de la etapa" es en concreto el nombre que corresponda en cada etapa.....	4
1.3. Validar que efectivamente se imprimen los mensajes de todas las etapas menos de 4, ¿cuáles? ¿Por qué?.....	6
1.4. Agregar un input con un contador que empiece en 0. El input debe estar bloqueado a escritura. Mediante un botón, cada vez que se pulse, se incrementará el contador.....	7
1.5. ¿Al pulsar el botón, conseguimos visualizar algún mensaje adicional de las etapas del ciclo de vida de Vue?.....	8
1.6. Si todavía queda alguna etapa sin imprimir su mensaje, provocar que se impriman. ¿Cómo lo has hecho? ¿Qué es lo que sucede?.....	9

# 1. Práctica con VueJS

## 1.1. Agregar un componente nuevo a la base que nos crea vue cli de forma automática.

Agregamos un nuevo componente con clic derecho sobre components → New file

En mi caso se va a llamar: MiComponente.vue



## 1.2. Sobre este componente agregar un console.log ("nombre de la etapa") en cada una de las etapas del ciclo de vida de la instancia Vue. Donde "nombre de la etapa" es en concreto el nombre que corresponda en cada etapa.

Crear la estructura del nuevo componente, le añado un subtítulo para saber cuando empieza y que se ejecuta bien en el index. Le añado los console.log que se piden

```
proyecto01 > src > components > ▼ MiComponente.vue > {} "MiComponente.vue"
 1  <template>
 2    <div id="mi-componente">
 3      <h3>{{ titulo }}</h3>
 4    </div>
 5  </template>
 6
 7  <script>
 8    export default {
 9      name: "MiComponente",
10
11      data() {
12        return {
13          titulo: "Este es mi nuevo componente",
14        };
15      },
16
17      beforeCreate() {
18        console.log("1 - beforeCreate");
19      },
20      created() { // Al crear los métodos, observadores y eventos
21        console.log("2 - created");
22      },
23      beforeMount() { // Se ejecuta antes de insertar el DOM
24        console.log("3 - beforeMount");
25      },
26      mounted() { // Se ejecuta al insertar el DOM
27        console.log("4 - mounted");
28      },
29      beforeUpdate() { // Se ejecuta al detectar un cambio
30        console.log("5 - beforeUpdate");
31      },
32      updated() { // Se ejecuta al realizar los cambios
33        console.log("6 - updated");
34      },
35      beforeDestroy() { // Se ejecuta antes de destruir la instancia
36        console.log("7 - beforeDestroy");
37      },
38      destroyed() { // Se ejecuta al destruir toda la instancia
39        console.log("8 - destroyed");
40      },
41    };
42  </script>
```

También hay que modificar la clase padre de los componentes: App.vue

Lo primero importo el componente, después indico dentro de export default que voy a usar el componente importado y por último en la etiqueta div indico que el nuevo componente se muestra debajo del titulo de “Vue pruebas”

```
<template>
  <div id="app">
    <HelloWorld msg="Vue pruebas" />
    <mi-componente></mi-componente>
  </div>
</template>

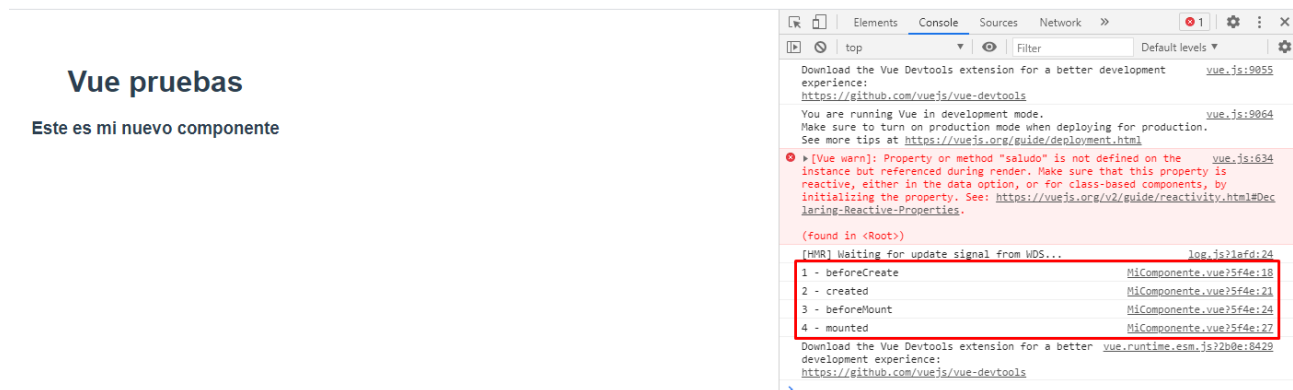
<script>
import HelloWorld from "../components/HelloWorld.vue";
import MiComponente from "../components/MiComponente.vue";

export default {
  name: "App",
  components: {
    HelloWorld,
    MiComponente,
  },
};
</script>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

### 1.3. Validar que efectivamente se imprimen los mensajes de todas las etapas menos de 4, ¿cuáles? ¿Por qué?

Desde el navegador → Clic derecho: inspeccionar → En la pestaña consola



Se observa las etapas que se han realizado, menos las últimas 4: beforeUpdate, updated, beforeDestroy y destroyed

- beforeUpdate y updated: Se ejecutan cuando detectan algún cambio en el HTML o en la instancia de Vue
- beforeDestroy y destroyed: Se ejecutan al destruir un elemento HTML o algún elemento en la instancia de Vue

#### 1.4. Agregar un input con un contador que empiece en 0. El input debe estar bloqueado a escritura. Mediante un botón, cada vez que se pulse, se incrementará el contador

Agrego un botón al componente y un input que diga el número de clicks que se han dado al botón

```
<template>
  <div id="mi-componente">
    <h3>{{ titulo }}</h3>
    <button v-on:click="counter += 1">Contador</button>
    <p>Se ha hecho clic en el botón de arriba {{ counter }} veces.</p>
  </div>
</template>

<script>
export default {
  name: "MiComponente",

  data() {
    return {
      titulo: "Este es mi nuevo componente",
      counter: 0
    };
  },
}
```

## 1.5. ¿Al pulsar el botón, conseguimos visualizar algún mensaje adicional de las etapas del ciclo de vida de Vue?

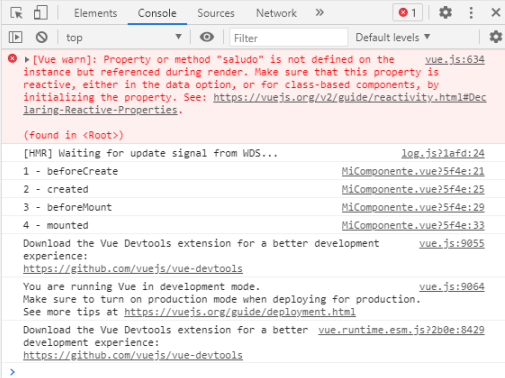
Como pasaba anteriormente se crean las 4 primeras etapas del ciclo de vida de Vue

### Vue pruebas

Este es mi nuevo componente

Contador

Se ha hecho clic en el botón de arriba 0 veces.



Ahora al darle al botón del contador se puede ver que se añadieron 2 nuevas etapas: beforeUpdate y updated

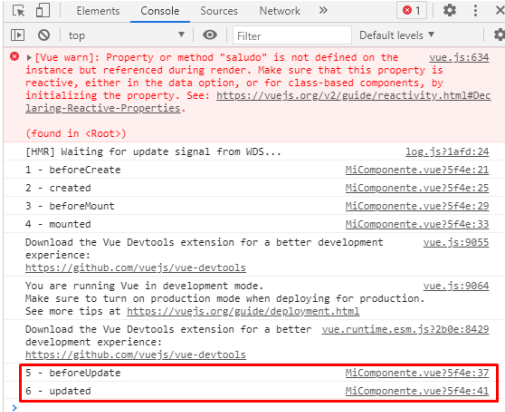
Esto se debe a que se ha ocurrido una modificación en el HTML

### Vue pruebas

Este es mi nuevo componente

Contador

Se ha hecho clic en el botón de arriba 1 veces.





## 1.6. Si todavía queda alguna etapa sin imprimir su mensaje, provocar que se impriman. ¿Cómo lo has hecho? ¿Qué es lo que sucede?

Para probar estas últimas dos etapas crea un botón que llama a una función y destruirá la instancia

El botón:

```
<template>
<div id="mi-componente">
  <h3>{{ titulo }}</h3>
  <button v-on:click="counter += 1">Contador</button>
  <button id="destroyButton" class="btn btn-danger" v-on:click="onDestroyClick">Destroy</button>
  <p>Se ha hecho clic en el botón de arriba {{ counter }} veces.</p>
</div>
</template>
```

El método:

```
methods: {
  onDestroyClick: function () {
    this.$destroy();
  },
}
```

Se pueden ver los eventos de: beforeDestroy y destroyed al pulsar el botón

### Vue pruebas

Este es mi nuevo componente

Contador Destroy

Se ha hecho clic en el botón de arriba 0 veces.

The screenshot shows the Vue DevTools Console with the following content:

- Warning: Property or method "saludo" is not defined on the instance but referenced during render. Make sure that this property is reactive, either in the data option, or for class-based components, by initializing the property. See: <https://vuejs.org/v2/guide/reactivity.html#Detecting-Reacting-Properties>. (found in <Root>)
- Download the Vue Devtools extension for a better development experience: <https://github.com/vuejs/vue-devtools>
- You are running Vue in development mode. Make sure to turn on production mode when deploying for production. See more tips at <https://vuejs.org/guide/deployment.html>
- [HMR] Waiting for update signal from WDS...
- 1 - beforeCreate
- 2 - created
- 3 - beforeMount
- 4 - mounted
- 7 - beforeDestroy
- 8 - destroyed