****

**VUE**

**Apuntes y Documentación**

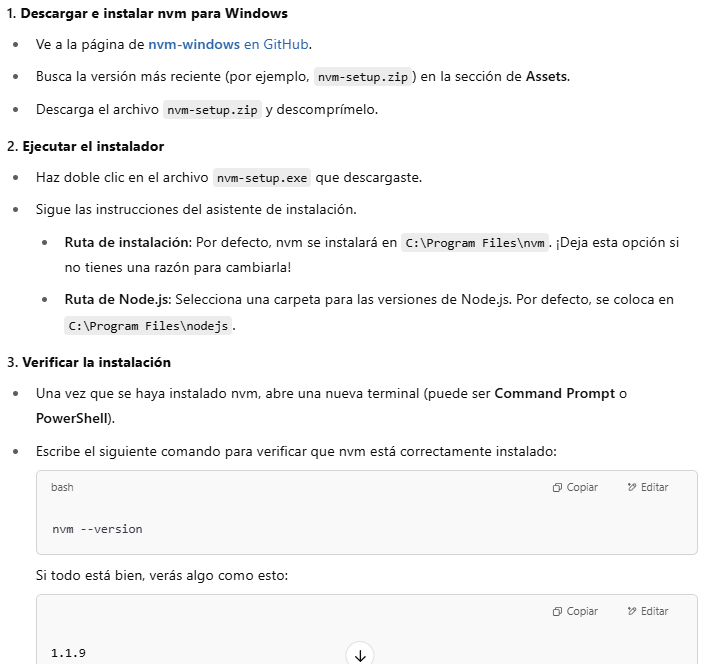
**+ INFORMACIÓN DE NODE**

**nvm ->** Node Version Manager.

**npm ->** Node Package Manager.

Para usar nvm para Windows hay que seguir estos pasos:

* **Descarga *“nvm-setup.zip”* desde:** <https://github.com/coreybutler/nvm-windows/releases>



* **Para instalar versiones:**

*nvm install 18.20.2*

*nvm install 14.8.0*

* **Para desinstalar versiones:**

*nvm uninstall 18.20.2*

*nvm uninstall 14.8.0*

* **Para cambiar de versión:**

*nvm use 18.20.2*

* **Para ver todas las versiones instaladas y la que se está usando:**

*nvm list*

* **Para cambiar todas las dependencias de un proyecto si no funcionan con esa versión de node:**

*rm -rf node\_modules*

*npm install*

**LA VERSIÓN DE NODE DE NUMIER ES LA 14.8.0**

**1 CARACTERÍSTICAS PRINCIPALES**

[**https://vuejs.org/guide/essentials/template-syntax.html**](https://vuejs.org/guide/essentials/template-syntax.html)

[**https://www.w3schools.com/vue/vue\_props.php**](https://www.w3schools.com/vue/vue_props.php)

**CREAR E INICIAR UN PROYECTO EN VUE:** [**https://vue3-spanish.netlify.app/guide/quick-start.html**](https://vue3-spanish.netlify.app/guide/quick-start.html)

**La característica principal de Vue es la representación declarativa : mediante una sintaxis de plantilla que extiende HTML, podemos describir cómo debería verse el HTML según el estado de JavaScript. Cuando el estado cambia, el HTML se actualiza automáticamente.**

***ref():*** *Toma el argumento y lo devuelve envuelto dentro de un objeto de referencia con una propiedad .value.*

*import { ref } from 'vue'*

*const count = ref(0)*

console.log(count) // { value: 0 }

console.log(count.value) // 0

count.value++

console.log(count.value) // 1

**CÓDIGO DE PRUEBA**

<script setup>

import { ref } from 'vue'

// component logic

// declare some reactive state here.

</script>

<template>

<h1>Make me dynamic!</h1>

</template>

# ***1.1- Acceso global restringido​***

Las expresiones de plantilla están protegidas y solo tienen acceso a una lista restringida de globales . Esta lista expone globales integradas de uso común, como ***Math*** y ***Date***.

Las variables globales que no se incluyan explícitamente en la lista, por ejemplo, las propiedades asociadas por el usuario en **[nombre window del archivo]**, no estarán disponibles en las expresiones de plantilla. Sin embargo, puede definir explícitamente variables globales adicionales para todas las expresiones de Vue añadiéndolas a ***[nombre del archivo] app.config.globalProperties***.

**2 SINTAXIS *DE PLANTILLA***

**Vue utiliza una sintaxis de plantilla (Template) basada en HTML que permite vincular declarativamente el DOM renderizado a los datos de la instancia del componente subyacente. Todas las plantillas de Vue son HTML sintácticamente válidas y pueden ser analizadas por navegadores y analizadores HTML compatibles con las especificaciones.**

# ***2.1- Interpolación de texto***

La forma más básica de enlace de datos es la interpolación de texto utilizando la sintaxis *“Mustache”* (llaves dobles):

<span>Message: {{ msg }}</span>

# ***2.2- HTML sin procesar***

La sintaxis *“Mustache”* interpreta los datos como texto plano, no como HTML. Para generar HTML real, necesitará usar la ***v-html*** directiva :

<p>Using text interpolation: {{ rawHtml }}</p>

<p>Using v-html directive: <span v-html="rawHtml"></span></p>

Usando interpolación de texto: <span style="color: red">Esto debería ser rojo.</span>

Usando la directiva v-html: Esto debe ser rojo.

# ***2.3- Enlaces de atributos***

No se puede usar *“Mustache”* dentro de atributos HTML. En su lugar, use una ***v-bind* directiva**:

<div v-bind:id="dynamicId"></div>

La v-bind directiva indica a Vue que mantenga el atributo id del elemento sincronizado con la propiedad *“dynamicId”* del componente. Si el valor enlazado es *null* o *undefined*, el atributo se eliminará del elemento renderizado.

## ***Sintaxis abreviada***

<div :id="dynamicId"></div>

## ***Sintaxis del mismo nombre***

Si el atributo tiene el mismo nombre que el valor de JavaScript que se va a vincular, la sintaxis se puede acortar aún más para omitir el valor del atributo:

<!-- same as :id="id" -->

<div :id></div>

<!-- this also works -->

<div v-bind:id></div>

# ***2.4- Atributos booleanos***

Los atributos booleanos son atributos que pueden indicar valores verdaderos o falsos según su presencia en un elemento. Por ejemplo, *“disabled”* es uno de los atributos booleanos más utilizados.

v-bind funciona un poco diferente en este caso:

<button :disabled="isButtonDisabled">Button</button>

El atributo disabled se incluirá si *“isButtonDisabled”* tiene un valor verdadero. También se incluirá si el valor es una cadena vacía, manteniendo la coherencia con *<button disabled=" ">*. Para otros valores falsos, el atributo se omitirá.

# ***2.5- Vinculación dinámica de múltiples atributos***

Si tiene un objeto JavaScript que representa múltiples atributos que se ve así:

const objectOfAttrs = {

id: 'container',

class: 'wrapper',

style: 'background-color:green'

}

Puedes vincularlos a un solo elemento usando v-bind sin un argumento:

<div v-bind="objectOfAttrs"></div>

# ***2.6- Uso de expresiones de JavaScript***

## ***Sólo expresiones en “Moustache”***

Un enlace con sintaxis *“Moustache”* sólo puede contener una expresión, si intentamos introducir una expresión condicional dentro, no funcionaría, por lo que tendremos que utilizar un ternario.

❌ **MAL**

**—-------------------------**

{{ var a = 1 }}

✅**BIEN**

**—-------------------------**

var a = 1

{{ a }}

❌ **MAL**

**—-------------------------**

{{ if (ok) { return message } }}

✅**BIEN**

**—-------------------------**

{{ ok ? message : “Nada” }}

## ***Funciones de llamada***

Es posible llamar a un método expuesto por un componente dentro de una expresión de enlace:

<time :title="toTitleDate(date)" :datetime="date">

{{ formatDate(date) }}

</time>

# ***2.7- Directivas***

## 

Las directivas son atributos especiales con el prefijo **v-**. Vue proporciona varias directivas integradas , entre ellas, **v-html** y **v-bind**, que presentamos anteriormente.

Se espera que los valores de los atributos de directiva sean expresiones JavaScript individuales (con la excepción de **v-for**, **v-on** y **v-slot**, que se tratarán en sus respectivas secciones más adelante). La función de una directiva es aplicar actualizaciones reactivas al DOM cuando cambia el valor de su expresión. **v-if** Por ejemplo:

<p v-if="seen">Now you see me</p>

Aquí, **v-if** eliminaría o insertaría el elemento <p> en función de la veracidad del valor de la expresión *“seen”*.

## ***Argumentos***

Algunas directivas pueden aceptar un *“argumento”*, indicado por dos puntos después del nombre de la directiva. Por ejemplo, la v-bind directiva se utiliza para actualizar reactivamente un atributo HTML:

const url = ref('title')

<a v-bind:href=”url”> ... </a>

<!-- Sintaxis abreviada -->

<a :href=”url”> ... </a>