



Escuela de Vue

El framework progresivo de JavaScript

Sergio Azócar

Frontend Developer



Vue Lover



Gamer



Geek



Neurodivergente

sergioazocar.com

@sergioazoc



The background features a large, semi-transparent Vue.js logo, which is a stylized 'V' composed of two overlapping triangles. The background is dark blue and decorated with various horizontal bars in white, pink, and cyan. The text is centered over the logo.

The Progressive JavaScript Framework



Evan You

Creator de Vue, Vite...

Ecosistema

Vue

Vue Router

Pinia

Vite

Vitest

Conceptos Básicos

- Single File Components

```
<template>  
  ...  
</template>  
  
<script>  
  ...  
</script>  
  
<style>  
  ...  
</style>
```



Conceptos Básicos

Options API

```
<script>
export default {
  // Properties returned from data() becomes reactive state
  // and will be exposed on `this`.
  data() {
    return {
      count: 0
    }
  },

  // Methods are functions that mutate state and trigger updates.
  // They can be bound as event listeners in templates.
  methods: {
    increment() {
      this.count++
    }
  },

  // Lifecycle hooks are called at different stages
  // of a component's lifecycle.
  // This function will be called when the component is mounted.
  mounted() {
    console.log(`The initial count is ${this.count}.`)
  }
}
</script>

<template>
  <button @click="increment">count is: {{ count }}</button>
</template>
```

Composition API

```
<script>
import { ref, onMounted } from 'vue'

export default {
  setup() {
    // reactive state
    const count = ref(0)

    // functions that mutate state and trigger updates
    function increment() {
      count.value++
    }

    // lifecycle hooks
    onMounted(() => {
      console.log(`The initial count is ${count.value}.`)
    })

    return {
      count,
      increment
    }
  }
}
</script>

<template>
  <button @click="increment">Count is: {{ count }}</button>
</template>
```

Composition API setup

```
<script setup>
import { ref, onMounted } from 'vue'

// reactive state
const count = ref(0)

// functions that mutate state and trigger updates
function increment() {
  count.value++
}

// lifecycle hooks
onMounted(() => {
  console.log(`The initial count is ${count.value}.`)
})
</script>

<template>
  <button @click="increment">Count is: {{ count }}</button>
</template>
```

Conceptos Básicos

- Interpolation

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

template

```
{{ number + 1 }}
```

```
{{ ok ? 'YES' : 'NO' }}
```

```
{{ message.split('').reverse().join('') }}
```


Conceptos Básicos

- Attributes binding

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

template

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

template

```
<span :title="toTitleDate(date)">
  {{ formatDate(date) }}
</span>
```

template

Conceptos Básicos

– Directives

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

template

v-text

v-html

v-show

v-if

v-else

v-else-if

v-for

v-on

v-bind

v-model

v-slot

v-pre

v-once

v-memo

v-cloak

Conceptos Básicos

– Reactive

```
<script setup>
import { reactive } from 'vue'

const state = reactive({ count: 0 })

function increment() {
  state.count++
}
</script>

<template>
  <button @click="increment">
    {{ state.count }}
  </button>
</template>
```

vue

* No funciona con primitivos (string, number ó boolean)

Conceptos Básicos

– Ref

```
<script setup>
import { ref } from 'vue'

const count = ref(0)

function increment() {
  count.value++
}
</script>

<template>
  <button @click="increment">
    {{ count }} <!-- no .value needed -->
  </button>
</template>
```

* Acepta cualquier cosa

Conceptos Básicos

– Computed

```
<script setup>
import { reactive, computed } from 'vue'

const author = reactive({
  name: 'John Doe',
  books: [
    'Vue 2 - Advanced Guide',
    'Vue 3 - Basic Guide',
    'Vue 4 - The Mystery'
  ]
})

// a computed ref
const publishedBooksMessage = computed(() => {
  return author.books.length > 0 ? 'Yes' : 'No'
})
</script>

<template>
  <p>Has published books:</p>
  <span>{{ publishedBooksMessage }}</span>
</template>
```

vue

Conceptos Básicos

- Class binding

```
const isActive = ref(true)  
const hasError = ref(false)
```

js

```
<div  
  class="static"  
  :class="{ active: isActive, 'text-danger': hasError }"  
></div>
```

template

resultado renderizado

```
<div class="static active"></div>
```

template

Conceptos Básicos

- Style binding


```
const activeColor = ref('red')  
const fontSize = ref(30)
```

js

```
<div :style="{ color: activeColor, fontSize: fontSize + 'px' }"></div>
```

template

resultado renderizado

```
<div style="color: red; font-size: 30px;"></div>
```

Conceptos Básicos

- Composables

```
const useUtils = () => {  
  const randomId = () => crypto.randomUUID()  
  
  return {  
    randomId,  
  }  
}  
  
export default useUtils
```

```
import useUtils from '@composables/useUtils'  
  
const { randomId } = useUtils()  
  
const newId = randomId()
```


Conceptos Básicos

– Components

```
<!-- BlogPost.vue -->
<script setup>
defineProps(['title'])
</script>

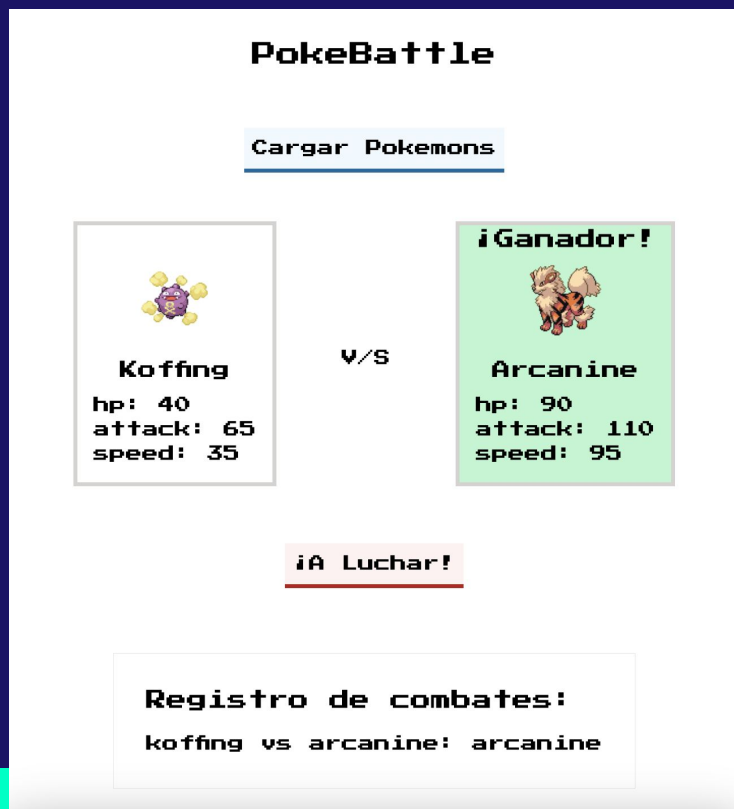
<template>
  <h4>{{ title }}</h4>
</template>
```

vue

```
<BlogPost title="My journey with Vue" />
<BlogPost title="Blogging with Vue" />
<BlogPost title="Why Vue is so fun" />
```

template

Qué vamos a construir?



Un registro de batallas
Pokémon!

Requisitos

01

Node v18+
(npm, yarn, pnpm)

02

Editor de texto
(VSCode)

03

Conocimientos en
JavaScript/Vue 3

04

Cuenta en Github



```
curl_easy_setopt(comm, CURLOPT_URL, url);  
if (curl_easy_getinfo(comm, CURLINFO_HTTP_CODE, &ihttpcode) != CURLE_OK)  
    fprintf(stderr, "Failed to set URL [%s]\n", errorcode);  
else;  
    curl_easy_setopt(comm, CURLOPT_FOLLOWLOCATION,  
        1);  
curl_easy_setopt(comm, CURLOPT_WRITEFUNCTION,  
        write_callback);  
code = curl_easy_perform(comm);  
if (code != CURLE_OK)  
    fprintf(stderr, "Failed to set redirect option [%s]\n",  
        errorcode);  
else;  
    curl_easy_setopt(comm, CURLOPT_WRITEFUNCTION,  
        write_callback);  
code = curl_easy_perform(comm);  
if (code != CURLE_OK)  
    fprintf(stderr, "Failed to set writer [%s]\n",  
        errorcode);
```

LET'S CODE!

