

Practica VUE 02 - 02 - Creando Componentes

Componentes vue

Puntos importantes en vue.

- Una aplicación vue esta formada por componentes. Componente inicial es el componente App que se guarda en el fichero App.vue es el inicial, y desde el siempre se importan los otros componentes.
- Un Componentes vue es el elemento Básico de Vue definen una etiqueta que podemos usar en la parte template otro componente vue
- Cada Componentes vue es un fichero con este formato NombreComoponente.vue
- Cada Componentes vue tiene su Contenido tres partes

```
<template> <h1>Hola Mundo {{ algo }}</h1> </template>
<script> export default { data() {return { algo: 'Soy el algo ' }} } </script>
<style> h1 { color: blue; } </style>
```

- Siempre tienen que tener una etiqueta padre en template

Ejemplo 1 componente (etiquetas)

Cada componente es una etiqueta que se inserta en la parte template de otro componente en este caso <MiPrimerComponente />

```
<template>
  <MiPrimerComponente />
</template>
```

podemos poner propiedades que con la sintaxis {{ }}

```
<template>
  <h1>Este es mi componente</h1> <br>
  <h2>{{ subtitle }}</h2>
</template>
```

Crea un componente que dentro del <h1> pon “Este es mi primer componente” y debajo un <h2> donde llame a un subtítulo definido en el script.

En App.vue

```
<script setup>
  import MiPrimerComponente from './components/MiPrimerComponente.vue';
</script>

<template>
  <MiPrimerComponente />
</template>
```

En MiPrimerComponente.vue

```
<template>
<div>
  <h1>Este es mi componente</h1> <br>
  <h2>{{ subtítulo }}</h2>
</div>
</template>

<script>
  export default {
    data() {
      return {
        subtítulo: 'Soy el subtítulo del componente'
      }
    }
  }
</script>
```

Ejemplo 2 Componentes compuestos y Props (atributos)

Cada componente es una etiqueta los props son los atributos de la etiqueta y se pueden usar otros componentes dentro de otro componente usando .

los props se definen dentro de un componente la parte script en el siguiente formato

```
<script>
export default {
    props: {
        nombre: { type: String, required: true},
        edad: { type: Number, required: true },
        estado: { type: String, required: true }
    }
}
</script>
```

la forma de insertar {{ cuenta }}

Para usar un componente (etiqueta) hijo dentro de otro componente padre se debe declarar con **import** dentro de la parte de script del padre y si hay props se debe usar **export**

```
import PropsHijo from './PropsHijo.vue';
export default { components: { PropsHijo } }
```

Crea 2 componente donde uno sea el padre y el otro el hijo,en el hijo irán las definiciones de las propiedades (nombre, edad, estado) y en el padre el valor.

En PropsPadre.vue

```
<template>
    <h1>
        <PropsHijo nombre="Jordi Hurtado" edad="109" estado="inmortal" />
    </h1>
</template>

<script>
    import PropsHijo from './PropsHijo.vue';
    export default { components: { PropsHijo } }
</script>
<style>
    h1 { color: blue; }
</style>
```

En PropsHijo.vue

```
<template>
  <div>
    <p>Mi nombre es {{ nombre }}</p>
    <p>Mi edad es {{ edad }} años</p>
    <p>Soy {{ estado }} </p>
  </div>
</template>

<script>
export default {
  props: {
    nombre: { type: String, required: true},
    edad: { type: Number, required: true },
    estado: { type: String, required: true }
  }
}
</script>
```

En App.vue

```
<script setup>
  import PropsPadre from './components/PropsPadre.vue';
</script>

<template>
  <PropsPadre />
</template>
```

Ejercicios

1. De que esta hecha una aplicacion vue?
2. Componente inicial una aplicacion vue? donde se guarda ?
3. Que define un Componentes vue? en que parte se usa ?
4. Como creamos Componentes vue?
5. partes de un Componentes vue? Explicalas
6. Que significa que un componente solo tiene una etiqueta padre?

7. Explica que es un props
8. Como se pueden usar un componente dentro de otro componente
9. Haz el ejemplo 1
10. Haz el ejemplo 2
11. Modifica el ejemplo 2 para poner estilos de color rojo el h1
12. Modifica el ejemplo 1 para añadir un cuarto **props** llamado programa que salga junto a los otros y añade estilos a los anteriores para que cada elemento salga de un color diferente.
13. Modifica el ejemplo 1 para añadir un segundo componente PropsHijo2 que tenga tres props cabecera1 cabecera2 cabecera3 y que cada uno de los props pasados se pongan en estilo h1 .. h3 cada uno con un color distinto debe salir algo parecido a esto

```
propshijo1
```

Mi nombre es Jordi Hurtado

Mi edad es 109 años

Soy inmortal

trabajo en el programa Saber y ganar

```
Propshijo2
```

Cabecera1 Esto es la cabecera 1

Cabecera2 Esto es la cabecera 2

Cabecera3 Esto es la cabecera 1

Practica VUE 03 - Metodos y eventos en Vue

Ejemplo 3 (Metodos de los componentes)

Hay dos formas de dotar de comportamiento a los objetos dentro de la sección **export default** de **<script>** mediante la siguiente sintaxis

```
<script>
    export default {
        data() { return { cuenta: 0 } },
        methods: { incrementar() {this.cuenta++; } },
        computed: { doble() { return this.cuenta * 2; } }
    }
</script>
```

el metodo **data** devuelve los valores iniciales **data() { return { cuenta: 0 } },**

la sección **methods** define los eventos que se pueden llamar con **@click**

```
methods: { incrementar() {this.cuenta++; } },
```

la sección **computed** define los calculos privados que realiza el objeto

```
computed: { doble() { return this.cuenta * 2; } }
```

Crea un botón que cada vez que pulses lo cuente y que al mismo tiempo te calcule el doble.

En ContadorBoton.vue

```
<template>
<button @click="incrementar">
    El contador es: {{ cuenta }}, el doble es: {{ doble }}
</button>
</template>

<script>
    export default {
        data() { return { cuenta: 0 } },
        methods: {
            incrementar() {this.cuenta++; },
            computed: {
                doble() { return this.cuenta * 2; }
            }
        }
    }
</script>

<style>
button { font-size: 2em; border: 1px solid black; background-color: aqua; padding: 10px; }
```



```
</style>
```

En App.vue

```
<script setup>
import ContadorBoton from './components/ContadorBoton.vue'
</script>
<template>
<ContadorBoton />
</template>
```

Ejemplo 4 (Pasando parámetros)

Podemos pasar parametros uniendo los props y los metodos pero hay que tener cuidado que no modifiquemos ningun props (esto es es los props son solo de entrada) si no da un error de *props no mutables*

Crea un botón que cada vez que pulses lo cuente y que al mismo tiempo te calcule el doble y que se pueda pasar el valor inicial mediante un atributo valor

En ContadorBoton.vue

```
<template>
<button @click="incrementar">
    El contador es: {{ cuenta }}, el doble es: {{ doble }}
</button>
</template>

<script>
    export default {
        data() { return { cuenta: 0 } },
        methods: {
            incrementar() {this.cuenta++},
        },
        computed: {
            doble() { return this.cuenta * 2; }
        }
    }
</script>

<style>
button { font-size: 2em; border: 1px solid black; background-color: aqua; padding: 10px; }
</style>
```

En App.vue

```
<script setup>
import ContadorBoton from './components/ContadorBoton.vue'
</script>
<template>
<ContadorBoton />
</template>
```

Ejercicios

1. Haz el ejemplo 3
2. Haz el ejemplo 4
3. Modifica el ejemplo 4 para cambiar los colores de todos los valores con style
4. Modifica el ejemplo 4 para añadir los siguientes valores adicionales triple, cuadruple, cuadrado, cubo añadiendo las funciones compute correspondientes
5. Modifica el ejemplo 4 para añadir los eventos reiniciar y decrementar debe quedar algo así
6. Modifica el ejemplo 4 para añadir un segundo **props** que se llame incremento que sea cuanto se incrementa al pulsar en los botones . debe quedar algo así

incrementar
decrementar
reiniciar

El contador se inicia en valor : 0

El incremento es : 4

la variable cuenta es 60

el doble es: 120

el triple es: 180

el cuadruple es: 240

el cuadrado es: 3600

el cubo es: 216000