



Unidad # 1

Desarrollo Frontend con librerías del lado del cliente

Programación Computacional IV

Gisela Espinoza



**UNIVERSIDAD
GERARDO BARRIOS**
Líderes en Gestión del Conocimiento





Competencia de la asignatura

Aplicar los lenguajes de programación y tecnologías para la Web con el propósito de utilizarlos en el desarrollo de aplicaciones basándose en estándares internacionales de desarrollo de software trabajando de forma individual y colaborativa.

Contenidos a desarrollar

Comunicación entre Componentes: propiedades y eventos



¿Comunicación entre Componentes: propiedades y eventos?





Pasando datos mediante métodos



Envío de datos entre componentes

En Vue.js se envían datos desde un componente padre hacia el hijo mediante sus propiedades (props) y desde el componente hijo hacia el padre mediante eventos (\$emit).

Para el ejemplo se crean dos componentes el primero captura los datos de los productos y el segundo componente mostrara el total de acuerdo con el precio y cantidad establecidos en el primer componente.

Envío de datos entre componentes

El primer componente será hermano de otro componente que capturará la información

Entonces en el HTML en donde estarán estos componentes será el siguiente:

```
Vue.component("producttotal",{
  ...
  props: ['total'],
  template: `
    <p>
      Total: {{ total }}
    </p>
  `
});
```

```
<div id="app">
  <!--llamado al componente-->
  <product
    v-on:changeTotal="updateTotal"></product>

  <productTotal
    :total="totalP"
  ></productTotal>
</div>
```

Envío de datos entre componentes

productTotal: solo recibe parámetros.

Product: estará escuchando el evento *changetotal* y en ese caso ejecutará la función *updateTotal* que actualizará el valor de *total*.

```
var vue=new Vue({
  el:"#app",
  data: {
    totalP:0
  },
  methods:{
    updateTotal: function(total){
      this.totalP=total;
    }
  }
});
```


Envío de datos entre componentes

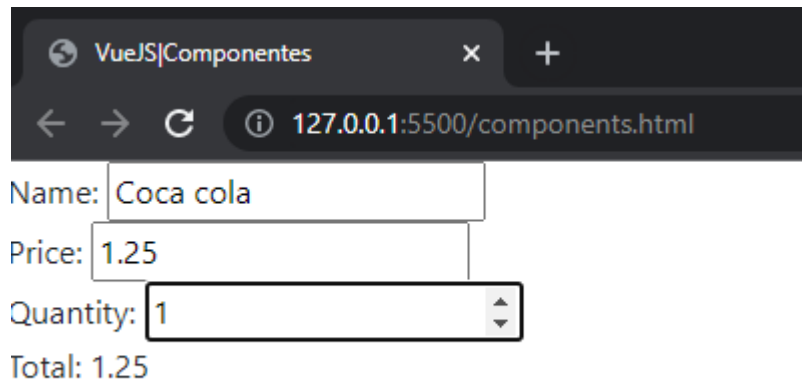
Por su parte, el componente *product* deberá emitir el evento respectivo con los valores necesarios para la función *updateTotal*.

```
Vue.component("product", {
  data: function() {
    return {
      name: "",
      price: 0,
      quantity: 0
    }
  },
  template: `
    <form>
      Name:
      <input type="text" v-model="name"/><br/>
      Price:
      <input type="text" v-model="price"/><br/>
      Quantity:
      <input type="number" v-model.number="quantity"/>
    </form>
  `,
  computed: {
    totalP: function() {
      {
        return this.price * this.quantity;
      }
    }
  },
  watch: {
    quantity: function() {
      {
        this.$emit('changetotal', this.totalP )
      }
    }
  },
  created: function() {
    {
      this.$emit('changetotal', this.totalP)
    }
  }
});
```

Envío de datos entre componentes

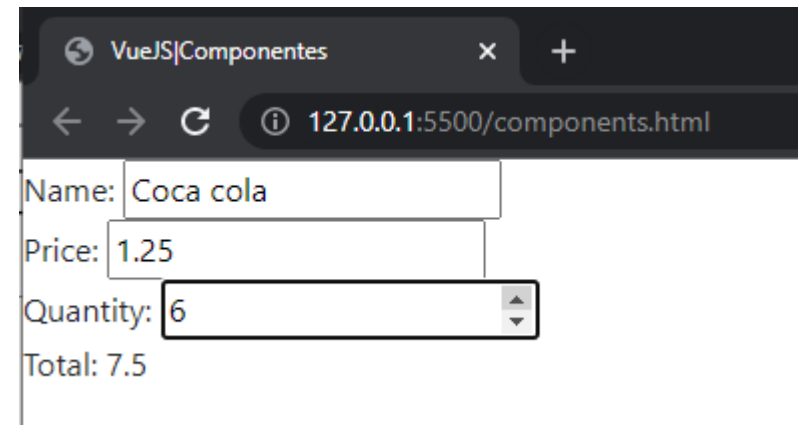
Ejecución:

Al cambiar la cantidad de producto, el total será actualizado.



A screenshot of a web browser window titled 'VueJS|Componentes'. The address bar shows '127.0.0.1:5500/components.html'. The form contains the following fields:

- Name: Coca cola
- Price: 1.25
- Quantity: 1 (with a spinner control)
- Total: 1.25



A screenshot of the same web browser window after the quantity has been changed. The form now displays:

- Name: Coca cola
- Price: 1.25
- Quantity: 6 (with a spinner control)
- Total: 7.5

Pasando datos mediante eventos



Envío de datos entre componentes

Paso de datos con el bus de eventos

Un bus de eventos no es más que otra instancia de Vue. Una instancia que puede tener las mismas propiedades que cualquier otra instancia. Se realizará un cambio en el ejemplo de la semana anterior se va a crear una instancia para generar en ella el evento *changetotal*.

Envío de datos entre componentes

Agregar las siguientes líneas

```
$ component.js > ...  
1   var bus = new Vue({});  
2
```

El componente *product* deberá emitir eventos sobre esta instancia y no sobre su misma instancia.

```
watch:  
{  
  quantity:function()  
  {  
    bus.$emit('changetotal',this.totalP )  
  }  
},  
created:function()  
{  
  bus.$emit('changetotal', this.totalP)  
}
```

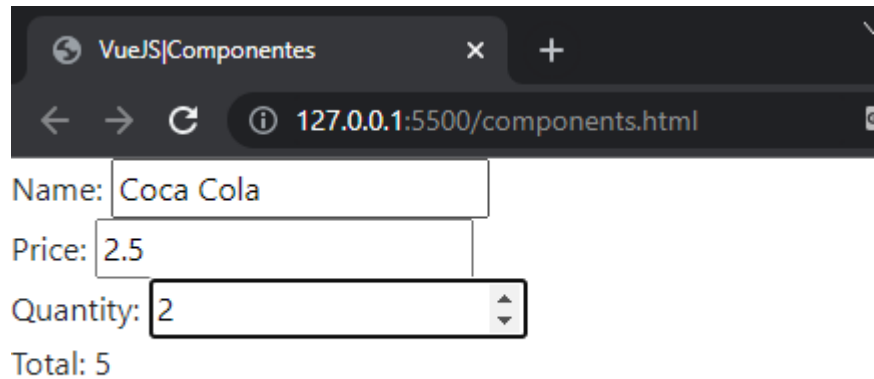
Envío de datos entre componentes

Como el bus es una instancia diferente no tiene mucho sentido conservar el evento en el HTML, por lo cual se simplificaría de la siguiente manera:

```
var vue=new Vue({  
  el:"#app",  
  data: {  
    totalP:0  
  },  
  methods:{  
    updateTotal: function(total){  
      this.totalP=total;  
    }  
  },  
  created: function(){  
    bus.$on('changetotal', this.updateTotal)  
  }  
});
```

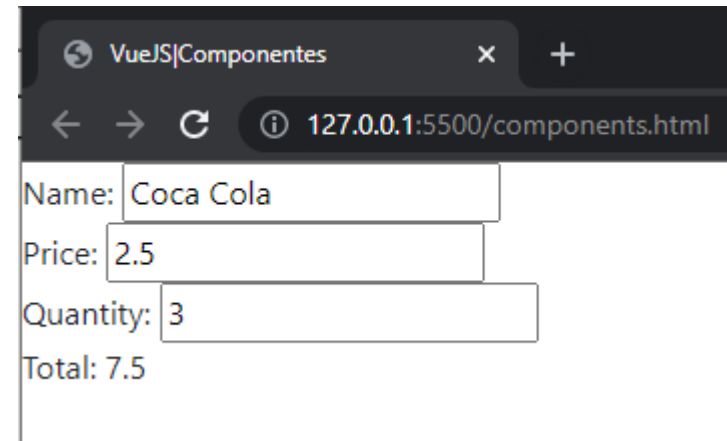
Envío de datos entre componentes

Ejecución



A screenshot of a web browser window titled 'VueJS|Componentes' with the address bar showing '127.0.0.1:5500/components.html'. The page displays a form with the following fields and values:

Field	Value
Name:	Coca Cola
Price:	2.5
Quantity:	2
Total:	5



A screenshot of a web browser window titled 'VueJS|Componentes' with the address bar showing '127.0.0.1:5500/components.html'. The page displays a form with the following fields and values:

Field	Value
Name:	Coca Cola
Price:	2.5
Quantity:	3
Total:	7.5

Tabla de actividades

Nombre de la actividad	Vuejs
Tipo de actividad	Repaso
Tipo de participación	Individual
Competencias específica de la asignatura	Aplicar los lenguajes de programación y tecnologías para la Web con el propósito de utilizarlos en el desarrollo de aplicaciones basándose en estándares internacionales de desarrollo de software trabajando de forma individual y colaborativa.
Instrucciones	Desarrolle cada ejercicio propuesto en clases y preséntelo.
Fecha de entrega	La fecha límite de participación será el domingo al final de la Semana, a las 11:59 p.m.
Instrumento de evaluación	Rúbrica de evaluación
Ponderación	Evaluación formativa

Recursos Complementarios

Recurso	Título	Cita Referencial
Documentación Oficial	Componente y propiedades en vue.js	https://es.vuejs.org/v2/guide/components.html#Pasando-datos-a-componentes-secundarios-con-Props

¿Preguntas?

¡Muchas gracias!