



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

- Eventos
- Manejadores de eventos
- Modificadores de eventos
- Clases y estilos en tiempo real





¿Qué son los eventos?

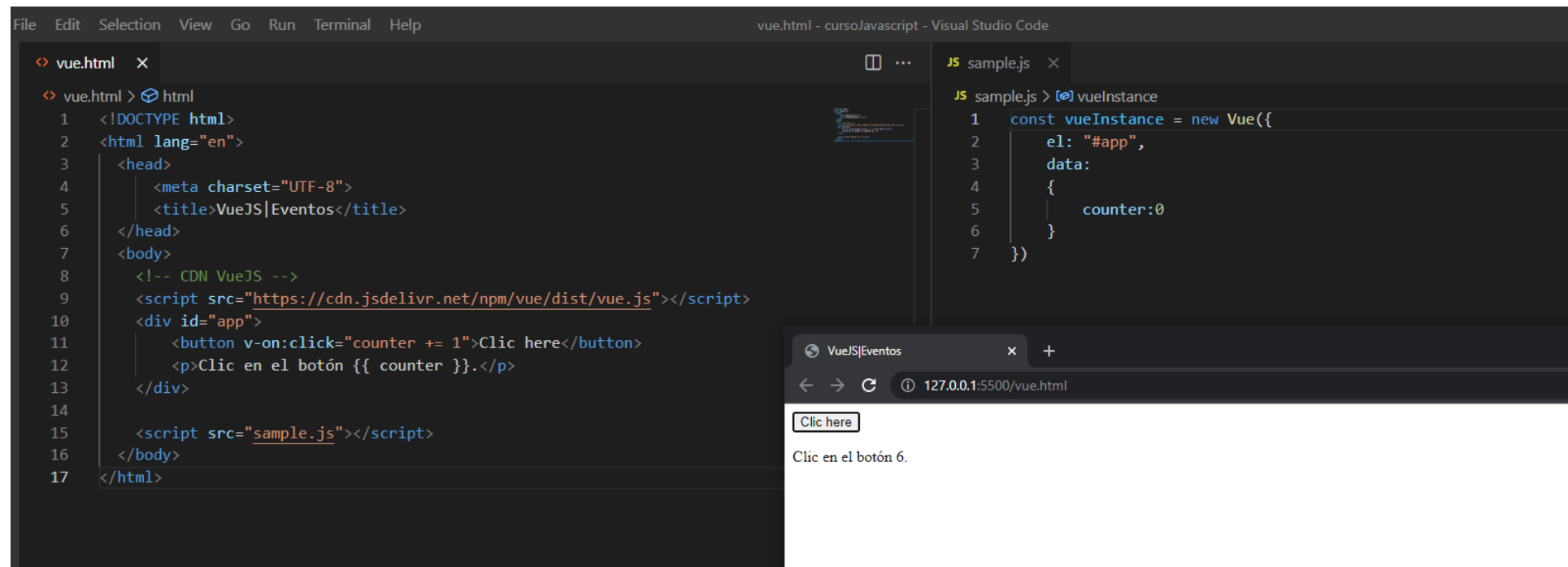


¿Que son los eventos?

- En JavaScript, la interacción con el usuario se consigue mediante la captura de los eventos que éste produce. Un evento es una acción del usuario ante la cual puede realizarse algún proceso (por ejemplo, el cambio del valor de un formulario, o la pulsación de un enlace).
- Los eventos se capturan mediante los manejadores de eventos. El proceso a realizar se programa mediante funciones JavaScript llamadas por los manejadores de eventos.

Manejadores de eventos en vue.js

Se puede utilizar la directiva v-on para escuchar eventos del DOM



The image shows a development environment with Visual Studio Code and a web browser. In Visual Studio Code, the file `vue.html` contains HTML code with a Vue.js script and a `sample.js` file containing a Vue instance. The browser displays the rendered page with a button and a paragraph.

```
File Edit Selection View Go Run Terminal Help vue.html - cursoJavascript - Visual Studio Code
```

```
< vue.html x
< vue.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>VueJS|Eventos</title>
6   </head>
7   <body>
8     <!-- CDN VueJS -->
9     <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
10    <div id="app">
11      <button v-on:click="counter += 1">Clic here</button>
12      <p>Clic en el botón {{ counter }}.</p>
13    </div>
14
15    <script src="sample.js"></script>
16  </body>
17 </html>
```

```
JS sample.js x
JS sample.js > [0] vueInstance
1 const vueInstance = new Vue({
2   el: "#app",
3   data:
4     {
5       counter:0
6     }
7 })
```

VueJS|Eventos x +
127.0.0.1:5500/vue.html
Clic here
Clic en el botón 6.

Manejadores de eventos en vue.js

La lógica para muchos controladores puede resultar mas compleja por lo que no es posible mantener el valor del atributo v-on, en este caso v-on también puede aceptar el nombre del método al que quiere llamar.

The image shows a code editor with two files: `vue.html` and `sample.js`.

vue.html content:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>VueJS|Eventos</title>
6   </head>
7   <body>
8     <!-- CDN VueJS -->
9     <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
10    <div id="app">
11      <!--button v-on:click="counter += 1">Clic here</button>
12      <p>Clic en el botón {{ counter }}.</p-->
13      <button v-on:click="greeting">Clic here</button>
14    </div>
15
16    <script src="sample.js"></script>
17  </body>
18 </html>
```

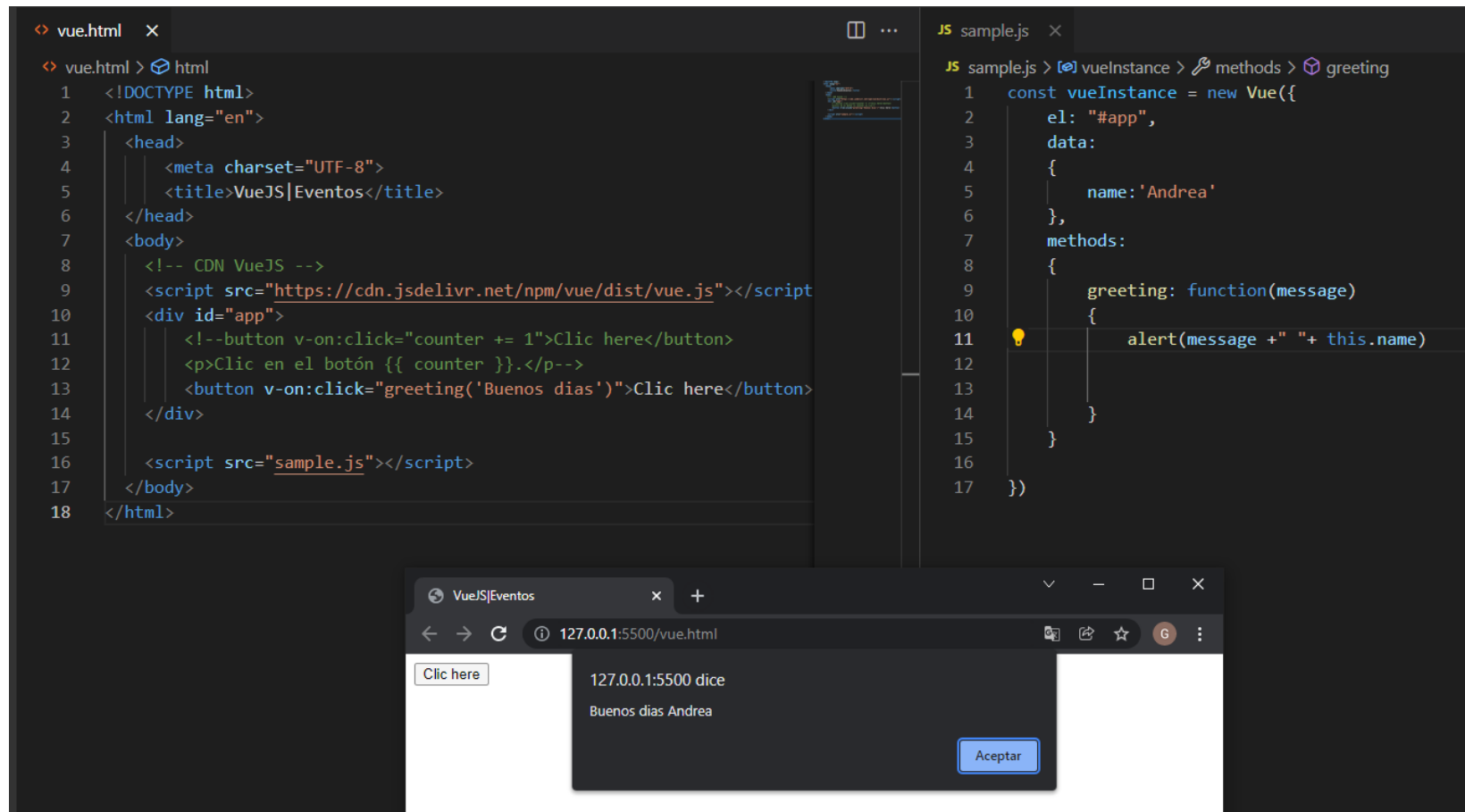
sample.js content:

```
1 const vueInstance = new Vue({
2   el: "#app",
3   data:
4     {
5       name: 'Andrea'
6     },
7   methods:
8     {
9       greeting: function(event)
10      {
11        alert('Buen día ' + this.name)
12
13        //evento del DOM nativo
14        if(event)
15        {
16          alert(event.target.tagName)
17        }
18      }
19    }
20 })
```

Below the code editor, a browser window titled "VueJS|Eventos" is shown at the URL `127.0.0.1:5500/vue.html`. It contains a button labeled "Clic here". An alert dialog box is displayed over the browser, showing the text "127.0.0.1:5500 dice Buen día Andrea" and an "Aceptar" button.

Manejadores en línea

En lugar de enlazar al nombre de un método, se puede utilizar métodos en una declaración JavaScript.



The image shows a code editor with two files: `vue.html` and `sample.js`. The `vue.html` file contains HTML code for a Vue.js application. It includes a `<script>` tag for Vue.js and a `<script>` tag for `sample.js`. The `sample.js` file contains a Vue.js instance with a `greeting` method. The browser preview shows a button labeled "Clic here" and an alert message "127.0.0.1:5500 dice Buenos dias Andrea".

```
vue.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>VueJS|Eventos</title>
6   </head>
7   <body>
8     <!-- CDN VueJS -->
9     <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
10    <div id="app">
11      <!--button v-on:click="counter += 1">Clic here</button>
12      <p>Clic en el botón {{ counter }}.</p-->
13      <button v-on:click="greeting('Buenos dias')">Clic here</button>
14    </div>
15    <script src="sample.js"></script>
16  </body>
17 </html>
```

```
sample.js > vueInstance > methods > greeting
1 const vueInstance = new Vue({
2   el: "#app",
3   data:
4     {
5       name: 'Andrea'
6     },
7   methods:
8     {
9       greeting: function(message)
10      {
11        alert(message + " " + this.name)
12      }
13    }
14 })
```

VueJS|Eventos
127.0.0.1:5500/vue.html
Clic here
127.0.0.1:5500 dice
Buenos dias Andrea
Aceptar

Modificadores de eventos

Vue.js proporciona modificadores de eventos para v-on para manejar detalles de eventos DOM, como: `event.preventDefault()` o `event.stopPropagation()`

Vue.js invoca el modificador mediante el sufijo de instrucción indicado por un punto (.)

`.stop`

`.prevent`

`.capture`

`.self`

`.once`

Modificadores de eventos

<!-- Se detendrá la propagación del evento click. --><a v-on:click.stop="hazEsto">

<!-- El evento de enviar ya no volverá a cargar la página. --><form v-on:submit.prevent="onSubmit"></form>

<!-- Los modificadores pueden encadenarse --><a v-on:click.stop.prevent="hazEsto">

<!-- solo el modificador --><form v-on:submit.prevent></form>

<!-- utilizar el modo de captura al agregar el detector de eventos -->

<!-- es decir, un evento dirigido a un elemento interno se maneja aquí antes de ser manejado por ese elemento -->

<div v-on:click.capture="hazEsto">...</div>

Modificadores de teclas

Cuando escuchamos eventos de teclado, a menudo necesitamos verificar códigos de teclas comunes. Recordar todos los keyCodes es una molestia, por lo que Vue proporciona alias para las claves más utilizadas:

```
.enter  
.tab  
.delete (captura las teclas "Eliminar" y "Retroceso")  
.esc  
.space  
.up  
.down  
.left  
.right
```

Clases y estilos



Clases y estilos reactivos

Uso de binding para obtener clases y estilos reactivos

Binding:class

Binding:style

Binding:class

Usando :class se puede definir clases al igual que con el atributo class pero aquí se usa comillas simples dentro de las comillas dobles, solo si se trata de un data se obvian las comillas simples.

```
<button :class = "'bg-danger'">Click here</button>
```

Si se desea aplicar varias clases a una etiqueta HTML:

```
<button :class = "['bg-warning', 'btn']">Click here</button>
```

Binding:style

Usando :style se puede cambiar el estilo de algún elemento HTML en tiempo de ejecución. Para esto se utiliza el nombre de la propiedad CSS seguido del valor que se quiere que tome entre comillas simples y todo dentro de {}.

```
<button :style = "{fontSize : num + 'px'}">Click here</button>  
<button :style = "{fontSize : '5px'}">Click here</button>
```

Si se quieren utilizar mas propiedades:

```
<button :style = "{color : value, background-color : 'white'}">Click here</button>
```

Binding:style

Vue también nos permite usar los atributos class y style junto con los binding :class y :style

```
<div class="bg-danger" :class="'p-3'">  
  <p>Texto.</p>  
</div>
```


Tabla de actividades

Nombre de la actividad	Eventos
Tipo de actividad	Capturador de evento de las teclas del teclado
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 una aplicación donde capture cada tecla que presiona en el teclado y agregue un color verde si es letra, rojo si es numérica, y amarillo si es otro tipo.
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 - Coevaluación
Ponderación	Evaluación formativa

Recursos Complementarios

Recurso	Título	Cita Referencial
Documentación Oficial	Eventos en vue.js	https://es.vuejs.org/v2/guide/events.html#Escuchar-eventos
Sitio Web	Estilos en Vue.js	https://jdonsan.gitbooks.io/desarrolla-aplicaciones-con-vuejs/content/estilos.html

¿Preguntas?

¡Muchas gracias!