

Introducción a JavaScript



¿Qué es JavaScript?

- JavaScript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web dinámica.
- Una página web dinámica es aquella que incorpora efectos, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.
- JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos.
- El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades.

¿Qué es JavaScript?

- A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems, como se puede ver en <http://www.sun.com/suntrademarks/>.

Breve historia

- A principios de los años 90 empezaban a desarrollarse las primeras aplicaciones web y por tanto, las páginas web comenzaban a incluir formularios complejos.
- Con unas aplicaciones web cada vez más complejas y una velocidad de navegación tan lenta, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario.
- **Brendan Eich**, un programador que trabajaba en Netscape, pensó que podría solucionar este problema...
- Curiosidad: *es también uno de los fundadores de Mozilla.*



Breve historia

- **Brendan Eich** adaptó otras tecnologías existentes (como *ScriptEase*) al navegador Netscape Navigator 2.0, que iba a lanzarse en 1995. Inicialmente, Eich denominó a su lenguaje *LiveScript*.
- Posteriormente, Netscape firmó una alianza con Sun Microsystems para el desarrollo del nuevo lenguaje de programación.
- Justo antes del lanzamiento Netscape decidió cambiar el nombre por el de JavaScript.
- ***La razón del cambio de nombre fue exclusivamente por marketing, ya que Java era la palabra de moda en el mundo informático y de Internet de la época.***



Breve historia

- La primera versión de JavaScript fue un completo éxito y Netscape Navigator 3.0 ya incorporaba la siguiente versión del lenguaje, la versión 1.1.
- Al mismo tiempo, Microsoft lanzó **JScript** con su navegador Internet Explorer 3.
- **JScript** era una copia de JavaScript al que le cambiaron el nombre para evitar problemas legales.



Internet Explorer 3.0



Breve Historia

- Para evitar una guerra de tecnologías, Netscape decidió que lo mejor sería estandarizar el lenguaje JavaScript.
- De esta forma, en 1997 se envió la especificación JavaScript 1.1 al organismo ECMA (*European Computer Manufacturers Association*).
- ECMA creó el comité TC39 con el objetivo de *"estandarizar de un lenguaje de script **multiplataforma e independiente** de cualquier empresa"*.
- El primer estándar que creó el comité TC39 se denominó **ECMA-262**, en el que se definió por primera vez el lenguaje **ECMAScript**.



Breve Historia

- Por este motivo, algunos programadores prefieren la denominación *ECMAScript* para referirse al lenguaje JavaScript.
- De hecho, JavaScript no es más que la implementación que realizó la empresa Netscape del estándar ECMAScript.
- En Junio de **1997** se publicó la primera edición del estándar ECMA-262.
- Un año después, en Junio **de 1998** se realizaron pequeñas modificaciones para adaptarlo al estandar ISO/IEC-16262 y se creó la **segunda edición**.
- La tercera edición del estándar ECMA-262 (*publicada en Diciembre de 1999*) es la versión que utilizan los navegadores actuales.

Incluir JavaScript en documentos XHTML

En el mismo
documento
XHTML

En un archivo
externo

En los
elementos
XHTML

Incluir JavaScript en el mismo documento XHTML

- El código JavaScript se encierra entre etiquetas `<script>` y se incluye en cualquier parte del documento. Aunque *es correcto incluir cualquier bloque de código en cualquier zona* de la página, se recomienda definir el código JavaScript dentro de la cabecera del documento (*dentro de la etiqueta `<head>`*):

`<head>`

```
<meta http-equiv="Content-Type" content="text/html;
```

```
  charset=iso-8859-1" />
```

```
<title>Ejemplo de código JavaScript en el propio documento</title>
```

```
<script type="text/javascript">
```

```
  alert("Un mensaje de prueba");
```

```
</script>
```

```
</head>
```

Ejemplo

Definir JavaScript en un archivo externo

- Las instrucciones JavaScript se pueden incluir en un archivo externo de tipo JavaScript que los documentos XHTML enlazan mediante la etiqueta `<script>`.
- Se pueden crear todos los archivos JavaScript que sean necesarios y cada documento XHTML puede enlazar tantos archivos JavaScript como necesite.

```
<script type="text/javascript" src="js/codigo.js"></script>
```

- Archivo `codigo.js`:

```
alert("Un mensaje de prueba");
```

Ejemplo

Incluir JavaScript en los elementos XHTML

- Este último método es el menos utilizado, ya que consiste en incluir trozos de JavaScript dentro del código XHTML de la página:

```
<body>  
  <p>Haga click  
  <span onclick="alert('Un mensaje de prueba')">  
    acá.  
  </p>  
</body>
```

Ejemplo

El mayor inconveniente de este método es que *ensucia* innecesariamente el código XHTML de la página y complica el mantenimiento del código JavaScript. En general, este método sólo se utiliza para definir algunos eventos y en algunos otros casos especiales, como se verá más adelante.

Etiqueta noscript

- El lenguaje HTML define la etiqueta `<noscript>` para mostrar un mensaje al usuario cuando su navegador no puede ejecutar JavaScript o no lo tiene activado.

```
<body>
```

```
  <noscript>
```

```
    <p>Bienvenido a Mi Sitio</p>
```

```
    <p>La página que estás viendo requiere para su funcionamiento el  
uso de JavaScript. Si lo has deshabilitado intencionadamente, por favor vuelve  
a activarlo.</p>
```

```
  </noscript>
```

```
</body>
```

Ejemplo

La etiqueta `<noscript>` se debe incluir en el interior de la etiqueta `<body>` (normalmente se incluye al principio de `<body>`). El mensaje que muestra `<noscript>` puede incluir cualquier elemento o etiqueta XHTML.

Sintaxis

- No se tienen en cuenta los espacios en blanco y las nuevas líneas
- Se distinguen las mayúsculas y minúsculas
- No se define el tipo de las variables
- No es necesario terminar cada sentencia con el carácter de punto y coma (;) *
- Se pueden incluir comentarios:
 - // Ejemplo de comentario de una sola línea
 - /*Ejemplo de comentario de varias líneas: */

Posibilidades y limitaciones

- JavaScript fue diseñado de forma que se ejecutara en un entorno muy limitado
- Los scripts tampoco pueden cerrar ventanas que no hayan abierto esos mismos scripts.
- Los scripts no pueden acceder a los archivos del ordenador del usuario (ni en modo lectura ni en modo escritura) y tampoco pueden leer o modificar las preferencias del navegador.
- Si la ejecución de un script dura demasiado tiempo (por ejemplo por un error de programación) el navegador informa al usuario de que un script está consumiendo demasiados recursos y le da la posibilidad de detener su ejecución.

JavaScript en otros entornos

- Herramientas como Adobe Acrobat permiten incluir código JavaScript en archivos PDF.
- Otras herramientas de Adobe como Flash y Flex utilizan ActionScript, un dialecto del mismo estándar de JavaScript.
- Photoshop permite realizar pequeños scripts mediante JavaScript y la versión 6 de Java incluye un nuevo paquete (denominado javax.script) que permite integrar ambos lenguajes.
- Por último, aplicaciones como [Yahoo Widgets](#) y el [Dashboard de Apple](#) utilizan JavaScript para programar sus widgets.

Tipos de variables

- **Numéricas**

- `var iva = 16; // variable tipo entero`

- `var total = 234.65; // variable tipo decimal`

- **Cadenas de texto**

- `var mensaje = "Bienvenido a nuestro sitio web";`

- `var nombreProducto = 'Producto ABC';`

- `var texto1 = "Una frase con 'comillas simples' dentro";`

- `var texto2 = 'Una frase con "comillas dobles" dentro';`

- `var texto1 = 'Una frase con \'comillas simples\' dentro';`

- `var texto2 = "Una frase con \"comillas dobles\" dentro";`

Ejemplo

Tipos de variables

- **Arrays**

- `var dia1 = "Lunes";`

- `var dia2 = "Martes";`

- ...

- `var dia7 = "Domingo";`

- `var dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];`

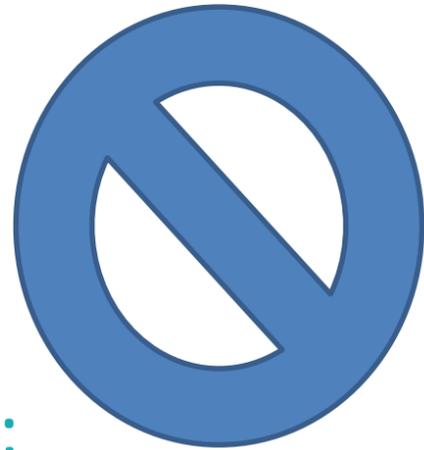
- `var diaSeleccionado = dias[0]; // diaSeleccionado = "Lunes"`

- `var otroDia = dias[5]; // otroDia = "Sábado"`

- **Booleanos**

- `var clienteRegistrado = false;`

- `var ivaIncluido = true;`



Ejemplo

Operadores

- **Asignación**
- `var numero1 = 3;`
- `numero1 = numero2;`
- **Incremento y decremento**
- `var numero = 5;`
- `++numero;`
- `--numero;`
- `numero1++;`
- `numero1--;`

Ejemplo

Operadores Lógicos

- **Negación**

- `var visible = true;`

- `alert(!visible);` // Muestra "false" y no "true"

- `var cantidad = 0;`

- `vacio = !cantidad;` // vacio = true

- `cantidad = 2;`

- `vacio = !cantidad;` // vacio = false

- `var mensaje = "";`

- `mensajeVacio = !mensaje;` // mensajeVacio = true

- `mensaje = "Bienvenido";`

- `mensajeVacio = !mensaje;` // mensajeVacio = false

Ejemplo

Operadores Lógicos

- **AND**
- La operación lógica AND obtiene su resultado combinando dos valores booleanos.
- El operador se indica mediante el símbolo && y su resultado solamente es true si los dos operandos son true:

& =ampersand

variable1	variable2	variable1 && variable2
true	true	true
true	false	false
false	true	false
false	false	false

Operadores Lógicos

- **OR**
- La operación lógica OR también combina dos valores booleanos.
- El operador se indica mediante el símbolo `||` y su resultado es `true` si alguno de los dos operandos es `true`:

variable1	variable2	variable1 variable2
true	true	true
true	false	true
false	true	true
false	false	false

Ejemplo

| = pipe

Operadores Matemáticos

- `var numero1 = 10;`
- `var numero2 = 5;`
- `resultado = numero1 / numero2; // resultado = 2`
- `resultado = 3 + numero1; // resultado = 13`
- `resultado = numero2 - 4; // resultado = 1`
- `resultado = numero1 * numero 2; // resultado = 50`
- `resultado = numero1 % numero2; // resultado = 0`
- `numero1 = 9;`
- `numero2 = 5;`
- `resultado = numero1 % numero2; // resultado = 4`

% → operador "*módulo*",
que calcula el resto de la
división entera de dos
números

Operadores Matemáticos

- **Los operadores matemáticos también se pueden combinar con el operador de asignación para abreviar su notación:**
- `var numero1 = 5;`
- `numero1 += 3; // numero1 = numero1 + 3 = 8`
- `numero1 -= 1; // numero1 = numero1 - 1 = 4`
- `numero1 *= 2; // numero1 = numero1 * 2 = 10`
- `numero1 /= 5; // numero1 = numero1 / 5 = 1`
- `numero1 %= 4; // numero1 = numero1 % 4 = 1`

Operadores Relacionales

- Los operadores relacionales definidos por JavaScript son idénticos a los que definen las matemáticas:
- mayor que ($>$),
- menor que ($<$),
- mayor o igual ($>=$),
- menor o igual ($<=$),
- igual que ($==$) y
- distinto de ($!=$).
- El operador $==$ se utiliza para comparar el valor de dos variables, por lo que es muy diferente del operador $=$, que se utiliza para asignar un valor a una variable:

Operadores Relacionales

- `var numero1 = 3;`
- `var numero2 = 5;`
- `resultado = numero1 > numero2; // resultado = false`
- `resultado = numero1 < numero2; // resultado = true`
- `numero1 = 5;`
- `numero2 = 5;`
- `resultado = numero1 >= numero2; // resultado = true` `resultado = numero1 <= numero2; // resultado = true` `resultado = numero1 == numero2; // resultado = true` `resultado = numero1 != numero2; // resultado = false`

Operadores Relacionales

- // El operador "=" asigna valores
- `var numero1 = 5;`
- `resultado = numero1 = 3;` // numero1 = 3 y resultado = 3

Ejemplo

- // El operador "==" compara variables
- `var numero1 = 5;`
- `resultado = numero1 == 3;` // numero1 = 5 y resultado = false

Estructuras de control de flujo

- **Estructura if**

```
if(condicion) {  
  ... }  
}
```

```
var mostrarMensaje = true;  
if(mostrarMensaje) {  
  alert("Hola Mundo"); }  
}
```



```
var mostrarMensaje = true;  
if(mostrarMensaje == true) {  
  alert("Hola Mundo"); }  
}
```

Las comparaciones siempre se realizan con el operador ==, ya que el operador = solamente asigna valores:

Ejemplo

Estructuras de control de flujo

- Estructura if...else

```
if(condicion)
{... }
else { ... }
```

```
if(edad < 12)
{ alert("Todavía eres muy pequeño"); }
else if(edad < 19) { alert("Eres un
adolescente"); } else if(edad < 35) { alert("Aun
sigues siendo joven"); }
else { alert("Estás viejo 😊"); }
```

No es obligatorio que la combinación de estructuras if...else acabe con la instrucción else, ya que puede terminar con una instrucción de tipo else if().

Estructuras de control de flujo

- Estructura for

```
for(inicializacion; condicion; actualizacion)  
{ ... }
```

```
var mensaje = "Hola, estoy dentro de un bucle";  
for(var i = 0; i < 5; i++)  
{ alert(mensaje); }
```

```
var dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];  
for(var i=0; i<7; i++)  
{ alert(dias[i]); }
```

Ejemplo

Estructuras de control de flujo

- **Estructura for...in**

```
for(indice in array)  
{ ... }
```

```
var dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado",  
"Domingo"];  
for(i in dias)  
{ alert(dias[i]); }
```

Si se quieren recorrer todos los elementos que forman un array, la estructura for...in es la forma más eficiente de hacerlo,

Funciones y propiedades básicas de JavaScript

- **Funciones útiles para cadenas de texto**

`length`

```
var mensaje = "Hola Mundo";  
var numeroLetras = mensaje.length; // numeroLetras = 10
```

`+`

```
var mensaje1 = "Hola";  
var mensaje2 = " Mundo";  
var mensaje = mensaje1 + mensaje2; // mensaje = "Hola Mundo"
```

`concat()`

```
var mensaje1 = "Hola";  
var mensaje2 = mensaje1.concat(" Mundo"); // mensaje2 = "Hola Mundo"
```

Funciones y propiedades básicas de JavaScript

- Funciones útiles para cadenas de texto

```
var mensaje1 = "Hola";  
var mensaje2 = mensaje1.toUpperCase(); // mensaje2 = "HOLA"
```

```
var mensaje1 = "HoIA";  
var mensaje2 = mensaje1.toLowerCase(); // mensaje2 = "hola"
```

```
var mensaje = "Hola";  
var letra = mensaje.charAt(0); // letra = H  
letra = mensaje.charAt(2); // letra = l
```

Funciones y propiedades básicas de JavaScript

- **Funciones útiles para cadenas de texto**

indexOf(caracter)

```
var mensaje = "Holaaaa";  
var posicion = mensaje.indexOf('a'); // posicion = 3  
posicion = mensaje.indexOf('b'); // posicion = -1
```

lastIndexOf(caracter)

```
var mensaje = "Holaaa";  
var posicion = mensaje.lastIndexOf('a'); // posicion = 5  
posicion = mensaje.lastIndexOf('b'); // posicion = -1
```

Funciones y propiedades básicas de JavaScript

- **Funciones útiles para cadenas de texto**

substring(inicio, final)

```
var mensaje = "Hola Mundo";  
var porcion = mensaje.substring(2); // porcion = "la Mundo"  
porcion = mensaje.substring(5,7); // porcion = "Mun"  
porcion = mensaje.substring(7); // porcion = "ndo"  
porcion = mensaje.substring(1, 8); // porcion = "ola Mun"  
porcion = mensaje.substring(3, 4); // porcion = "a"  
porcion = mensaje.substring(5, 0); // porcion = "Hola "  
porcion = mensaje.substring(0, 5); // porcion = "Hola "
```

split(separador)

```
var mensaje = "Hola Mundo, soy una cadena de texto!"; var palabra = "Hola"  
var palabras = mensaje.split(" ");  
// palabras = ["Hola", "Mundo,", "soy", "una", "cadena", "de", "texto!"];  
var letras = palabra.split(""); // letras = ["H", "o", "l", "a"]
```

Funciones y propiedades básicas de JavaScript

- **Funciones útiles para arrays**

length

```
var vocales = ["a", "e", "i", "o", "u"];  
var numeroVocales = vocales.length; // numeroVocales = 5
```

join(separador)

```
var array = ["hola", "mundo"];  
var mensaje = array.join(""); // mensaje = "holamundo"  
mensaje = array.join(" "); // mensaje = "hola mundo"
```

pop()

```
var array = [1, 2, 3];  
var ultimo = array.pop();  
// ahora array = [1, 2], ultimo = 3
```

Funciones y propiedades básicas de JavaScript

- **Funciones útiles para arrays**

push()

```
var array = [1, 2, 3];  
array.push(4); // ahora array = [1, 2, 3, 4]
```

shift()

```
var array = [1, 2, 3];  
var primero = array.shift(); // ahora array = [2, 3], primero = 1
```

unshift()

```
var array = [1, 2, 3];  
array.unshift(0); // ahora array = [0, 1, 2, 3]
```

reverse()

```
var array = [1, 2, 3];  
array.reverse(); // ahora array = [3, 2, 1]
```

Funciones y propiedades básicas de JavaScript

- **Funciones útiles para números**

NaN

```
var numero1 = 0;  
var numero2 = 0;  
alert(numero1/numero2); // se muestra el valor NaN
```

isNaN()

```
var numero1 = 0;  
var numero2 = 0;  
if(isNaN(numero1/numero2))  
{ alert("La división no está definida para los números indicados"); }  
else { alert("La división es igual a => " + numero1/numero2); }
```

Funciones y propiedades básicas de JavaScript

- **Funciones útiles para números**

Infinity

```
var numero1 = 10;  
var numero2 = 0;  
alert(numero1/numero2); // se muestra el valor Infinity
```

toFixed(digitos)

```
var numero1 = 4564.34567;  
numero1.toFixed(2); // 4564.35  
numero1.toFixed(6); // 4564.345670  
numero1.toFixed(); // 4564
```

Funciones

- `function` nombre_funcion() {
 ...
}

```
function suma_y_muestra()  
{ resultado = numero1 + numero2;  
  alert("La suma es" + resultado); }
```

```
var resultado;  
var numero1 = prompt("Introduce un número: ");  
var numero2 = prompt("Introduce otro número: ");  
suma_y_muestra();
```

Ejemplo

Argumentos y valores de retorno

```
• function nombre_funcion(argumento1, argumento2, ...argumentoN) {  
    ...  
}
```

```
// Definición de la función  
function calculaPrecioTotal(precio) {  
    var impuestos = 1.16;  
    var gastosEnvio = 10;  
    var precioTotal = ( precio * impuestos ) + gastosEnvio;  
}
```

```
// Llamada a la función y guarda en una variable el resultado  
var precioTotal = calculaPrecioTotal(23.34);
```

Ejemplo

Ámbito de las variables

```
function creaMensaje() {  
  var mensaje = "Mensaje de prueba";  
  creaMensaje();  
  alert(mensaje);  
}
```

Al ejecutar el código anterior no se muestra ningún mensaje por pantalla. La razón es que la variable mensaje se ha definido dentro de la función creaMensaje() y por tanto, es una **variable local** que solamente está definida dentro de la función.

```
function creaMensaje() {  
  var mensaje = "Mensaje de prueba";  
  alert(mensaje);  
  creaMensaje();  
}
```

Ejemplo

Ámbito de las variables

```
var mensaje = "Mensaje de prueba";  
function muestraMensaje() {  
  alert(mensaje); }
```

La variable mensaje se ha definido fuera de cualquier función. Este tipo de variables automáticamente se transforman en variables globales y están disponibles en cualquier punto del programa (incluso dentro de cualquier función).

```
function creaMensaje() {  
  mensaje = "Mensaje de prueba";  
  creaMensaje(); alert(mensaje);  
}
```

Ejemplo

Dentro de la función sin usar "var=" la variable es considerada global

Sentencias break y continue

- permiten manipular el comportamiento normal de los bucles for para detener el bucle o para saltarse algunas repeticiones.
- Concretamente, la sentencia break permite terminar de forma abrupta un bucle y la sentencia continue permite saltarse algunas repeticiones del bucle.

Ejemplo

Otras estructuras de control

```
while(condicion) { ... }
```

```
do { ... } while(condicion);
```

```
switch(variable) {  
  case valor_1: ... break;  
  case valor_2: ... break;  
  ... case valor_n: ... break;  
  default: ... break; }
```



VUE.JS

- Vue JS como su nombre indica es un framework javascript, es decir, es un conjunto de herramientas y funciones que permiten desarrollar páginas web de una manera más cómoda. Vue nace con la necesidad de no tener que escribir tanto código javascript y sobre todo con la idea de ahorrar tiempo al programador.
- Vue fue creado por Evan You, ex-trabajador de Google, que decidió crear su propio framework en el año 2014. Desde entonces, Vue ha subido muchísimo de popularidad, gracias a su sencillez y a todo lo que puede ofrecer.

VUE.JS

- Vue es un framework open source de JavaScript, el cual nos permite construir interfaces de usuarios de una forma muy sencilla.
- Hoy en día es uno de los Frameworks web más populares, junto con Angular y React.
- Un componente Vue, en términos simples, es un elemento el cual se encapsula código reutilizable.
- VueJS se autodenomina como un framework progresivo.
- Si queremos hacer uso de Vue no será necesario instalar absolutamente nada.

VUE.JS

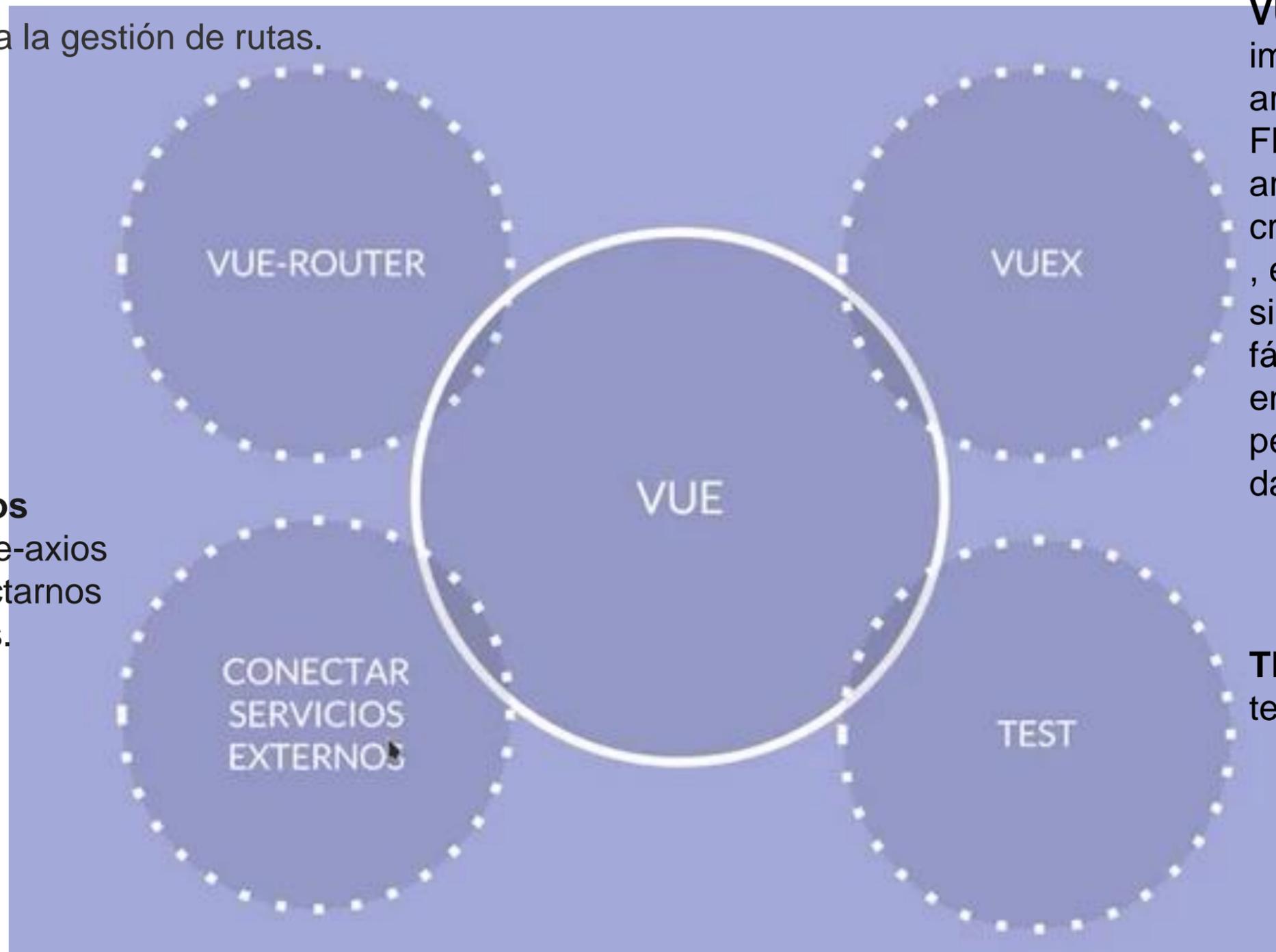
Características:

- Accesible.
- Versátil: Su núcleo es bastante pequeño y se escala a través de plugins, similar a React, una librería que cumple un propósito.
- Escalable por el mismo tema de la versatilidad.
- Reactivo.
- Optimizado: Su core ocupa 74Kb, lo que lo hace bastante liviano.
- Comunidad: Va creciendo a un ritmo importante con más 66500 estrellas en GitHub y 130 personas contribuyendo al core cada día.
- Licencia MIT: se publicó bajo el amparo de esta licencia.
- Una de las características más importantes de Vue es el trabajo con componentes.

VUE.JS - ECOSISTEMA

VUE-ROUTER: Para la gestión de rutas.

Conectar a servicios externos: Como vue-axios o vuefire para conectarnos a una base de datos.



VUEX: Es una implementación de la arquitectura de aplicación FLUX basada en la arquitectura ELM y además creado un poco en REDUX , es una implementación simple pero muy potente, fácil de utilizar y de entender, el cual nos permite gestionar el flujo de datos en nuestra aplicación.

TEST: Se pueden hacer test unitarios y funcionales.

MARCOS DE TRABAJO – VUE.JS - JAVASCRIPT

INSTALACION

The screenshot shows the Vue.js v2 guide page on es.vuejs.org. The page is in Spanish and provides instructions on how to install and use Vue.js. The main content area is titled "Instalación" and includes a section "Introducción" with sub-sections like "Qué es Vue.js?", "Empezando", "Renderización Declarativa", "Condicionales y Bucles", "Manejando Interacciones del Usuario", "Composición con Componentes", "Relación con Elementos Personalizados", "¿Listo para más?", and "La instancia Vue".

The main content area contains the following text:

La forma más fácil de probar Vue.js es usando [el ejemplo Hola Mundo en JSFiddle](#). Siéntase libre de abrirlo en otra pestaña y seguirlo mientras analizamos algunos ejemplos básicos. O puede [crear un archivo index.html](#) e incluir Vue con:

```
<!-- versión de desarrollo, incluye advertencias de ayuda en la consola -->
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
```

o con:

```
<!-- versión de producción, optimizada para tamaño y velocidad -->
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
```

La [página de instalación](#) proporciona más opciones de instalación de Vue. Nota: **No** recomendamos que los principiantes comiencen con `vue-cli`, especialmente si aún no están familiarizados con las herramientas de *build* basadas en Node.js.

Si prefiere algo más interactivo, también puede ver [esta serie de tutoriales en Scrimba](#), que le ofrece una combinación de *screencast* y *playground* de código con los que puede

On the right side, there are logos for Platinum sponsors: bit Share Code, tootwist, Vue School, VEHIKL, and NativeScript. At the bottom right, there is an advertisement for LiquidPlanner with the text: "Meet LiquidPlanner: The better way to align people, priorities, and projects. ads via Carbon".

Hola Mundo (1)

```
<> practica.html X JS objeto.js JS objeto2.js <> practica2.html
<> practica.html > body
1 <head>
2   <p>UTS - Marcos de Trabajo Segundo Corte</p>
3 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
4 <title>El primer script</title>
5 </head>
6 <body>
7   <div id="app">
8     <h1>{{titulo}}</h1>
9     <ul>
10    </ul>
11  </div>
12  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
13  <script type="text/javascript" src="objeto.js"></script>
14 </body>
15 </html>
```

```
JS objeto.js > [e] app
1  const app = new Vue({
2    el: '#app',
3    data: {
4      titulo: 'Hola Mundo',
5    }
6  })
```

UTS - Marcos de Trabajo Segundo Corte

Hola Mundo

Hola Mundo (2)

```
<> practica.html > body > div#app > input
1 <head>
2   <p>UTS - Marcos de Trabajo Segundo Corte</p>
3 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
4 <title>El primer script</title>
5 </head>
6 <body>
7   <div id="app">
8     <h1>{{titulo}}</h1>
9     <input v-model="titulo">
10  <ul>
11  </ul>
12  </div>
13  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
14  <script type="text/javascript" src="objeto.js"></script>
15 </body>
16 </html>
```

UTS - Marcos de Trabajo Segundo Corte

Hola Mundo Soy Victorrr

Hola Mundo Soy Victorrr

For Arrays (1)

```
<> practica.html > body > script
1 <head>
2   <p>UTS - Marcos de Trabajo Segundo Corte</p>
3 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
4 <title>El primer script</title>
5 </head>
6 <body>
7   <div id="app">
8     <h1>{{titulo}}</h1>
9     <ul>
10    <li v-for="equipo of equipos">{{equipo}}</li>
11  </ul>
12  </div>
13  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
14  <script type="text/javascript" src="objeto2.js"></script>
15 </body>
16 </html>
```

```
JS objeto2.js > ...
1 const app = new Vue({
2   el: '#app',
3   data: {
4     titulo: 'Marcos de Trabajo UTS',
5     equipos: ['Nacional', 'Bucaramanga', 'America']
6   }
7 })
```

Hola Mundo

Hola Mundo

- Nacional
- Bucaramanga
- America

For - Arrays (2)

```
JS objeto.js > [⌘] app
1  const app = new Vue({
2  el: '#app',
3  data: {
4    titulo: 'Marcos de Trabajo UTS',
5    equipos: [
6      {nombre:'Bucaramanga', Ciudad:'Medellin', Posicion:1},
7      {nombre:'Nacional',Ciudad:'Bucaramanga', Posicion:2},
8      {nombre:'America',Ciudad:'Cali', Posicion:3}
9    ]
10 }
11 })
```

Marcos de Trabajo UTS

Marcos de Trabajo UTS

- { "nombre": "Bucaramanga", "Ciudad": "Bucaramanga", "Posicion": 1 }
- { "nombre": "Nacional", "Ciudad": "Medellin", "Posicion": 2 }
- { "nombre": "America", "Ciudad": "Cali", "Posicion": 3 }

```
<ul>
|   <li v-for="equipo of equipos">{{equipo}}</li>
</ul>
```

For - Arrays (3)

```
JS objeto.js > [e] app > [k] data > [k] equipos
1  const app = new Vue({
2  el: '#app',
3  data: {
4    titulo: 'Marcos de Trabajo UTS',
5    equipos: [
6      {nombre: 'Nacional', Ciudad: 'Medellin', Posicion: 1},
7      {nombre: 'Bucaramanga', Ciudad: 'Bucaramanga', Posicion: 2},
8      {nombre: 'America', Ciudad: 'Cali', Posicion: 3}
9    ]
10 }
11 })
```

```
<ul>
  <li v-for="equipo of equipos">{{equipo}}</li>
</ul>
```

Marcos de Trabajo UTS

- { "nombre": "Nacional", "Ciudad": "Medellin", "Posicion": 1 }
- { "nombre": "Bucaramanga", "Ciudad": "Bucaramanga", "Posicion": 2 }
- { "nombre": "America", "Ciudad": "Cali", "Posicion": 3 }

```
<ul>
  <li v-for="equipo of equipos">{{equipo.nombre}}</li>
</ul>
```

Marcos de Trabajo UTS

- Nacional
- Bucaramanga
- America

For - Arrays (4)

```
<ul>
  <li v-for="equipo of equipos">
    {{equipo.nombre}} - {{equipo.Ciudad}}
  </li>
</ul>
```

Marcos de Trabajo UTS

Marcos de Trabajo UTS

- Bucaramanga - Bucaramanga
- Nacional - Medellin
- America - Cali

Marcos de Trabajo UTS

Marcos de Trabajo UTS

- Bucaramanga - Bucaramanga - 1
- Nacional - Medellin - 2
- America - Cali - 3

If

```
<ul>
  <li v-for="equipo of equipos">
    {{equipo.nombre}} - {{equipo.Ciudad}} - {{equipo.Posicion}}
    <span v-if="equipo.Posicion === 1" > - Campeon</span>
    <span v-else> - Eliminado</span>
  </li>
</ul>
```

Marcos de Trabajo UTS

Marcos de Trabajo UTS

- Bucaramanga - Bucaramanga - 1 - Campeon
- Nacional - Medellin - 2 - Eliminado
- America - Cali - 3 - Eliminado

Model – Conexión de Input con evento de procesamiento de datos

```
<body>
  <div id="app">
    <h1>{{titulo}}</h1>
    <input v-model='nuevequipo'>
    <button v-on:click="agregarequipo">Guardar</button>
    <ul>
      <li v-for="equipo of equipos">
        {{equipo.nombre}} - {{equipo.Ciudad}} - {{equipo.Posicion}}
        <span v-if="equipo.Posicion === 1" > - Campeon</span>
        <span v-else> - Eliminado</span>
      </li>
    </ul>
  </div>
```

```
JS objeto.js > [app] > methods > agregarequipo
1  const app = new Vue({
2  el: '#app',
3  data: {
4    titulo: 'Marcos de Trabajo UTS',
5    equipos: [
6      {nombre: 'Bucaramanga', Ciudad: 'Bucaramanga', Posicion:1},
7      {nombre: 'Nacional', Ciudad: 'Medellin', Posicion:2},
8      {nombre: 'America', Ciudad: 'Cali', Posicion:3}
9    ],
10   nuevoequipo: ''
11 },
12 methods:{
13   agregarequipo(){
14     console.log('Pasa por aqui');
15   }
16 }
17 }
18 )
```

UTS - Marcos de Trabajo Segundo Corte

Marcos de Trabajo UTS

- Bucaramanga - Bucaramanga - 1 - Campeon
- Nacional - Medellin - 2 - Eliminado
- America - Cali - 3 - Eliminado

Download the Vue Devtools extension for a better development experience: <https://github.com/vuejs/vue-devtools> vue.js:9055

You are running Vue in development mode. See more tips at <https://vuejs.org/guide/deployment.html> vue.js:9064

Pasa por aqui objeto.js:14

```
<input v-model='nuevequipo' v-on:keyup.enter="agregarequipo">
<button v-on:click="agregarequipo">Guardar</button>
```

Model – Conexión de Input con evento de procesamiento de datos

```

<body>
  <div id="app">
    <h1>{{titulo}}</h1>
    <input v-model='nueveequipo' v-on:keyup.enter="agregarequipo">
    <button v-on:click="agregarequipo">Guardar</button>
    <ul>
      <li v-for="equipo of equipos">
        {{equipo.Posicion}} - {{equipo.nombre}} -
        <input type="number" v-model.number="equipo.puntos">
        <span v-if="equipo.Posicion === 1" > - Campeon</span>
        <button v-on:click="equipo.puntos = equipo.puntos + 1">+</button>
      </li>
    </ul>
    <h4>TOTAL: {{SumarPuntos}}</h4>
  </div>
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  <script type="text/javascript" src="objeto.js"></script>
</body>

```

```

practica.html JS objeto.js X JS objeto2.js <> practica2.html
JS objeto.js > [?] app > computed > SumarPuntos
1  const app = new Vue({
2  el: '#app',
3  data: {
4    titulo: 'Marcos de Trabajo UTS',
5    equipos: [
6      {Posicion:1, nombre:'Bucaramanga', Ciudad:'Bucaramanga', puntos:9 },
7      {Posicion:2, nombre:'Nacional',Ciudad:'Medellin', puntos:7 },
8      {Posicion:3, nombre:'America',Ciudad:'Cali', puntos:4},
9      {Posicion:4, nombre:'Junior',Ciudad:'Cali', puntos:1}
10   ],
11   nuevoequipo: ''
12  },
13  methods:{
14    agregarequipo(){
15      this.equipos.push({
16        nombre: this.nuevoequipo, Ciudad:'Prueba', Posicion:4
17      })
18      this.nuevoequipo='';
19    }
20  },
21  computed:{
22    SumarPuntos(){
23      this.total=0;
24      for(equipos of this.equipos){
25        this.total=this.total + equipos.puntos
26      }
27      return this.total;
28    }
29  }
30 })

```

Marcos de Trabajo UTS

- 1 - Bucaramanga - - Campeon
- 2 - Nacional -
- 3 - America -
- 4 - Junior -

TOTAL: 39



Bootstrap

- Bootstrap es un framework CSS y Javascript diseñado para la creación de interfaces limpias y con un diseño responsive. Además, ofrece un amplio grupo de herramientas y funciones, de manera que los usuarios pueden crear prácticamente cualquier tipo de sitio web haciendo uso de los mismos.

Bootstrap

- Actualmente, Bootstrap es una de las alternativas más populares a la hora de desarrollar tanto sitios webs como aplicaciones. Una de las principales ventajas que ofrece es que permite la creación de sitios y apps 100% adaptables a cualquier tipo de dispositivo. Una cuestión de suma importancia teniendo en cuenta que a día de hoy son cada vez más los usuarios que acceden a Internet a través de sus teléfonos y tabletas.

Bootstrap

- Bootstrap es uno de los frameworks CSS de código abierto más conocidos y más utilizados en el mundo del desarrollo web.
- Este framework facilita enormemente la maquetación de páginas web, ya que te permite crear una interfaz muy limpia y completamente responsive, es decir, adaptable a cualquier tamaño de pantalla. Además, ofrece tantas herramientas y funcionalidades que te permite crear una web desde cero muy fácilmente.

Bootstrap

- Bootstrap fue desarrollado por Twitter y su principal función era la de dar forma a los sitios web mediante librerías CSS, pero poco tiempo después, concretamente en el año 2011, se liberó bajo la licencia MIT y en febrero de 2012 ya era el proyecto de desarrollo más popular de GitHub.

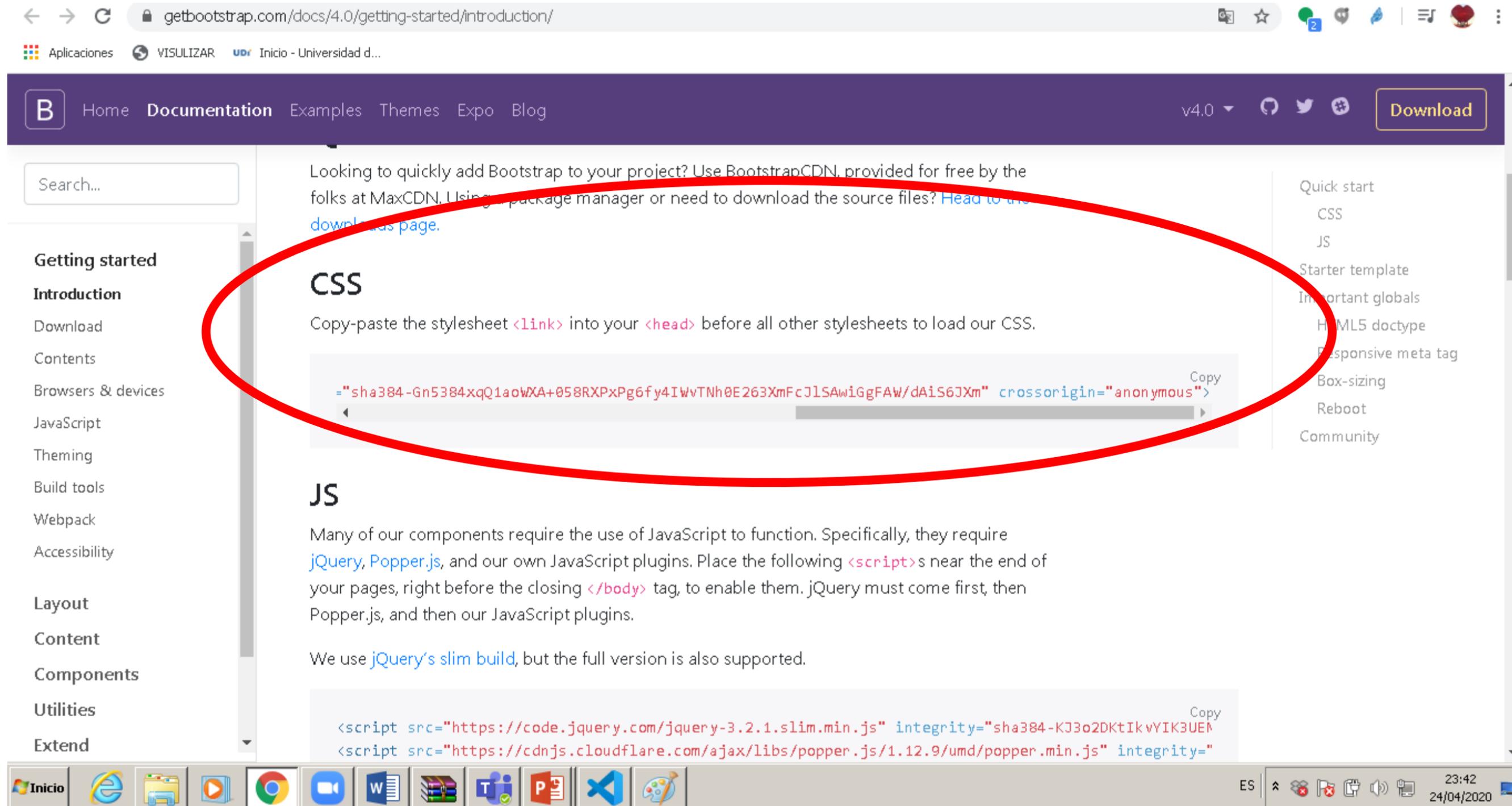
Bootstrap

- Bootstrap es compatible con la mayoría de navegadores web del mercado, y más desde la versión 3, actualmente es totalmente compatible con los siguientes navegadores:
- Google Chrome (en todas las plataformas).
- Safari (tanto en iOS como en Mac).
- Mozilla Firefox (en Mac y en Windows).
- Internet Explorer (en Windows y Windows Phone).
- Opera (en Windows y Mac).

Bootstrap - Ventajas

- Estas son las seis principales razones por las que Bootstrap es uno de los frameworks más utilizados:
- Soporte: Es una de las características más destacadas desde framework. Existe muchísima documentación en la red sobre su manejo y muchos blogs especializados en ello.
- Framework de código abierto: Esto quiere decir que todo su código está disponible y accesible a través de GitHub.
- Compatibilidad: No vas a tener problemas de incompatibilidad con los principales navegadores (Firefox, Google Chrome, Internet Explorer, Opera o Internet Explorer).
- Integración con librerías JavaScript: Tipografías, botones, cuadros...
- Facilidad de uso: Tanto a la hora de crear contenido como en el momento de subir archivos como imágenes o vídeos.
- Utiliza un sistema grid: Es decir, un sistema de cuadrículas (rejillas) que te permite crear el diseño de una web insertando el contenido en bloques o columnas.

Bootstrap - Instalación



Looking to quickly add Bootstrap to your project? Use BootstrapCDN, provided for free by the folks at MaxCDN. Using a package manager or need to download the source files? [Head to our download page.](#)

CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJLSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
```

JS

Many of our components require the use of JavaScript to function. Specifically, they require [jQuery](#), [Popper.js](#), and our own JavaScript plugins. Place the following `<script>`s near the end of your pages, right before the closing `</body>` tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.

We use [jQuery's slim build](#), but the full version is also supported.

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UEIEm82hGJ8hZJ4Eg50zBz9Mf8390j98" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-AzPg431nzVkPq46e9xgw6XFBCW5roVqmeBfE0z0qb4tIh2wN3EjY6O4EW4tob7e" crossorigin="anonymous"></script>
```

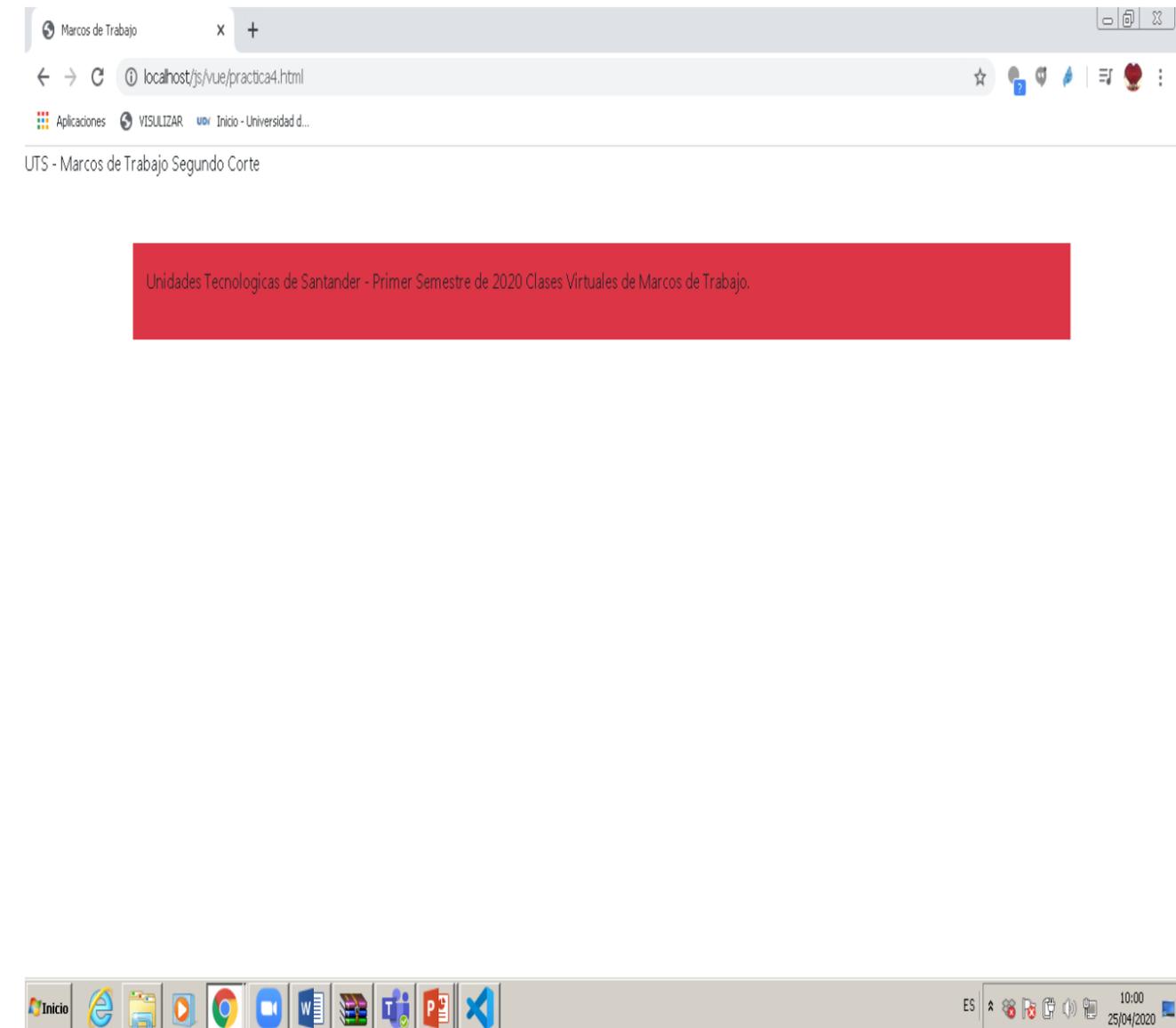
Bootstrap - Instalación

```
<> practica4.html > body > div#app
1 <head>
2   <p>UTS - Marcos de Trabajo Segundo Corte</p>
3 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
4 <title>Marcos de Trabajo</title>
5 <link rel="stylesheet"
6 href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
7 integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAi
8 crossorigin="anonymous">
9 </head>
10 <body>
11   <div id="app">
12     |
13   </div>
14   <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
15   <script type="text/javascript" src="objeto.js"></script>
16 </body>
17 </html>
```

```
JS objeto4.js > [e] app
1   const app = new Vue({
2     el: '#app',
3     data: {
4
5     },
6     methods: {
7
8     }
9   })
```

Bootstrap - Clases

```
> practica4.html > body > div#app.container.mt-5 > div.bg-danger
1 <head>
2   <p>UTS - Marcos de Trabajo Segundo Corte</p>
3 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
4 <title>Marcos de Trabajo</title>
5 <link rel="stylesheet"
6 href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
7 integrity="sha384-Gn5384xqQ1a0WXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAi
8 crossorigin="anonymous">
9 </head>
10 <body>
11   <div id="app" class="container mt-5">
12     <div class="bg-danger" :class="'p-3'">
13       <p>Unidades Tecnologicas de Santander - Primer Semestre de 2020
14         Clases Virtuales de Marcos de Trabajo.
15     </p>
16   </div>
17 </div>
18 <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
19 <script type="text/javascript" src="objeto.js"></script>
20 </body>
21 </html>
```



Bootstrap – Mas Clases

Unidades Tecnológicas de Santander - Primer Semestre de 2020 Clases Virtuales de Marcos de Trabajo.

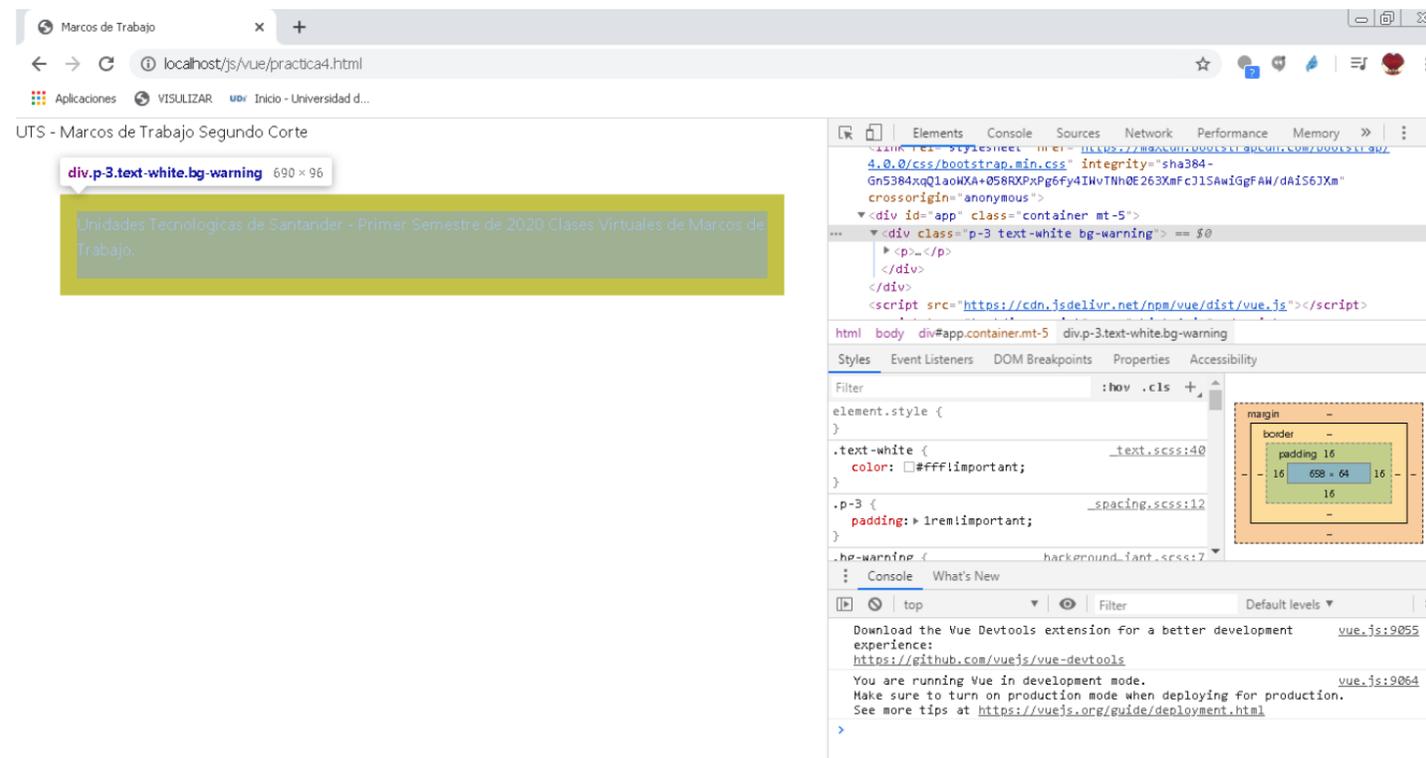
```
<div id="app" class="container mt-5">  
  <div class="bg-danger" :class="['p-3', 'text-white']">  
    <p>Unidades Tecnológicas de Santander - Primer Semestre de 2020  
      Clases Virtuales de Marcos de Trabajo.  
    </p>  
  </div>  
</div>
```

Bootstrap – Clases Dinamicas

```
<div id="app" class="container mt-5">  
  <div :class="['p-3', 'text-white', fondo]">
```

Unidades Tecnológicas de Santander - Primer Semestre de 2020 Clases Virtuales de Marcos de Trabajo.

```
JS objeto4.js > [🔗] app > 🛠 data  
1   const app = new Vue({  
2     el: '#app',  
3     data: {  
4       fondo="bg-danger"  
5     },  
6     methods: {  
7  
8   }  
9   })
```



Bootstrap – Clases Dinamicas

Array de
clases
especificas y
variables

```
<div id="app" class="container mt-5">  
  <div :class="['p-3', 'text-white', fondo]">  
    <p>Unidades Tecnologicas de Santander - Primer Semestre de 2020  
      Clases Virtuales de Marcos de Trabajo.  
    </p>  
  </div>  
  <input type="text" class="form-control my-3" v-model="fondo">  
</div>
```

Unidades Tecnologicas de Santander - Primer Semestre de 2020 Clases Virtuales de Marcos de Trabajo.

bg-warning

} danger
primary
dark

Bootstrap - Clases

```
<div id="app" class="container mt-5">
  <div :class="['p-3', 'text-white', fondo]">
    <p>Unidades Tecnológicas de Santander - Primer Semestre de 2020
      Clases Virtuales de Marcos de Trabajo.
    </p>
  </div>
  <input type="text" class="form-control my-3" v-model="fondo">
  <div class="text-white p-3" :class="{ 'bg-info': color, 'bg-success' :
!color}">
    <p>Quedate en casa</p>
  </div>
  <button class="btn btn-primary" @click="color = !color">Cambiar</button>
</div>
```

Bootstrap - Clases

```
<div id="app" class="container mt-5">
  <div :class="['p-3','text-white', fondo]">
    <p>Unidades Tecnológicas de Santander - Primer Semestre de 2020
      Clases Virtuales de Marcos de Trabajo.
    </p>
  </div>
  <input type="text" class="form-control my-3" v-model="fondo">
  <div class="text-white p-3" :class="{ 'bg-info': color, 'bg-success' :
    !color}">
    <p>Quedate en casa</p>
  </div>
  <button class="btn btn-primary" @click="color = !color">Cambiar</button>
</div>
```

```
JS objeto4.js > [?] app > methods
1  const app = new Vue({
2  el: '#app',
3  data: {
4    fondo: 'bg-warning',
5    color: false
6  },
7  methods: {}
8
9  }
10 }
```

UTS - Marcos de Trabajo Segundo Corte

Unidades Tecnológicas de Santander - Primer Semestre de 2020 Clases Virtuales de Marcos de Trabajo.

bg-warning

Quedate en casa

Cambiar

The screenshot shows the Chrome DevTools interface. The top part displays the DOM tree with the following structure:

```
<div id="app" class="container mt-5">
  <div class="p-3 text-white bg-warning">
    <p>Unidades Tecnológicas de Santander - Primer Semestre de 2020 Clases Virtuales de Marcos de Trabajo.
  </div>
  <input type="text" class="form-control my-3">
  <div class="text-white p-3 bg-info">
    <p>Quedate en casa
  </div>
  <button class="btn btn-primary">Cambiar
</div>
```

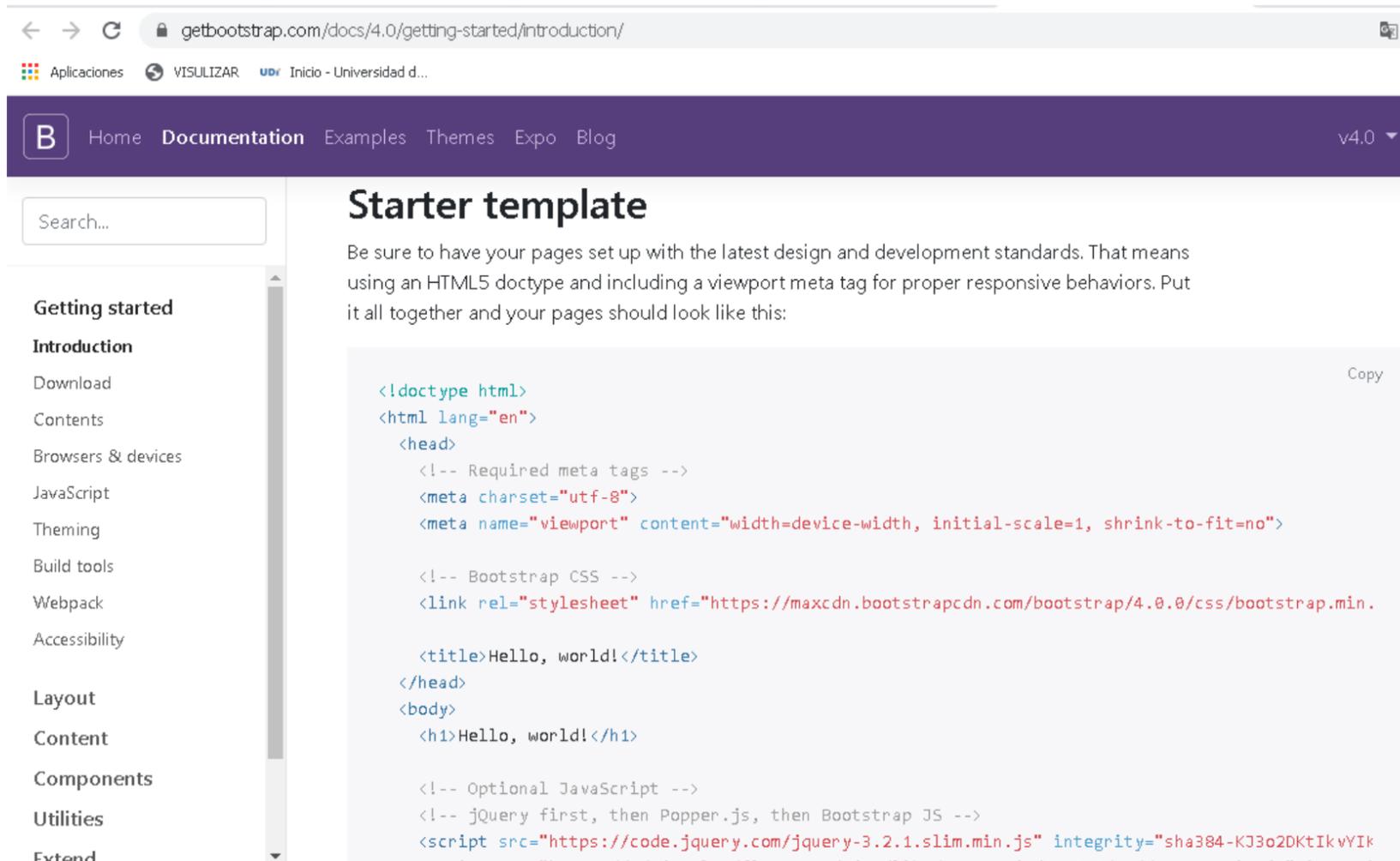
The Styles panel shows the following styles for the selected element:

```
element.style {
}
.text-white {
  color: #fff !important;
}
.p-3 {
  padding: 1rem !important;
}
.bg-warning {
  background-color: #ffc107;
}
```

The console shows the following messages:

```
Download the Vue Devtools extension for a better development experience:
https://github.com/vuejs/vue-devtools
You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html
```

CRUD - Create, Read, Update and Delete



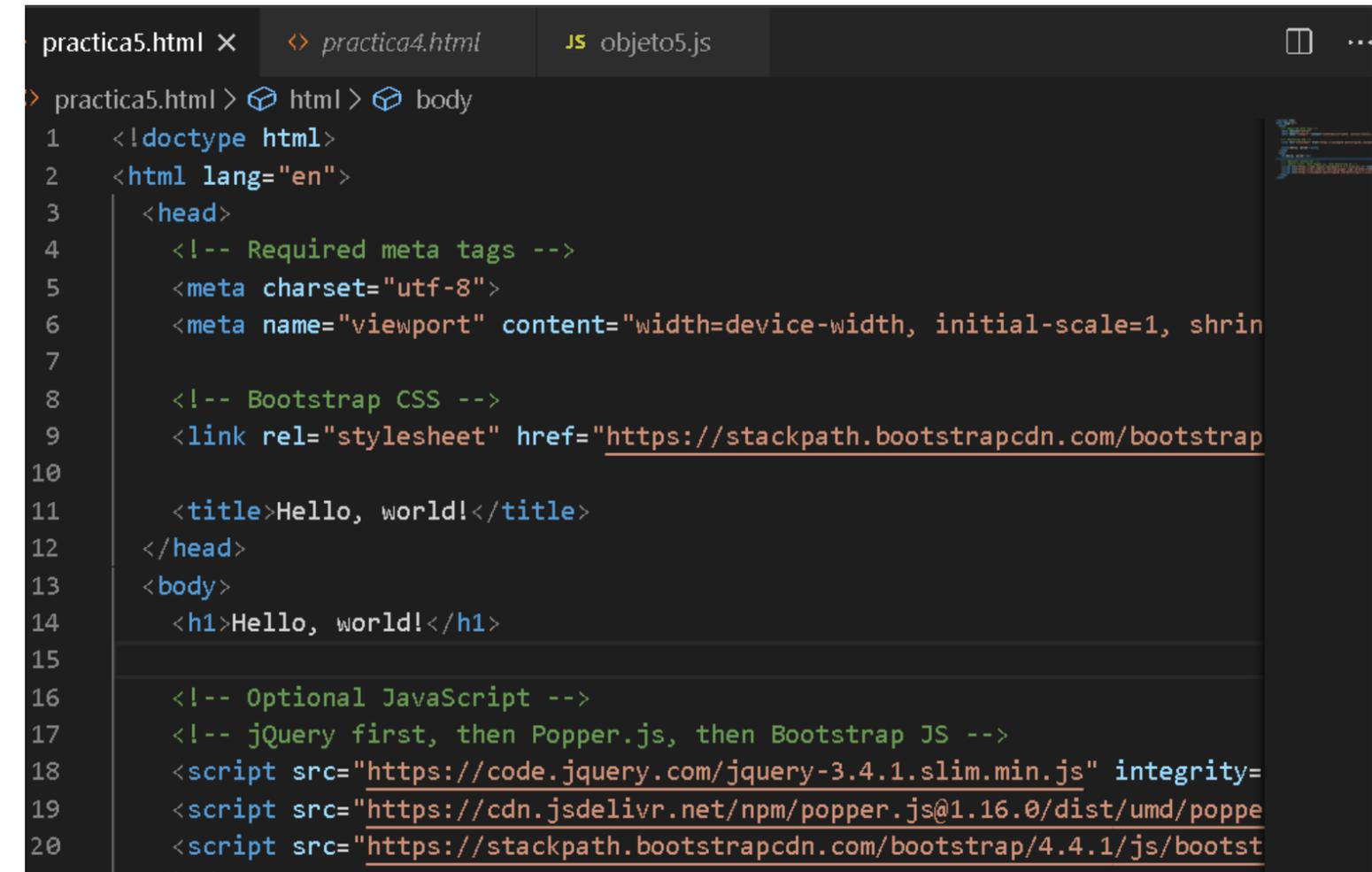
The screenshot shows the Bootstrap 4.0 documentation page. The browser address bar displays 'getbootstrap.com/docs/4.0/getting-started/introduction/'. The page title is 'Starter template'. The main content area contains the following HTML code:

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.

  <title>Hello, world!</title>
</head>
<body>
  <h1>Hello, world!</h1>

  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK
```



The screenshot shows a code editor with the following HTML code for 'practica5.html':

```
practica5.html > <> practica4.html JS objeto5.js
> practica5.html > html > body
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrin
7
8 <!-- Bootstrap CSS -->
9 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap
10
11 <title>Hello, world!</title>
12 </head>
13 <body>
14 <h1>Hello, world!</h1>
15
16 <!-- Optional JavaScript -->
17 <!-- jQuery first, then Popper.js, then Bootstrap JS -->
18 <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity=
19 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper
20 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootst
```

Quitamos esos JS y usamos los JS nuestros

CRUD - Create, Read, Update and Delete

```
> practica5.html > html > body
1  <!doctype html>
2  <html lang="en">
3    <head>
4      <!-- Required meta tags -->
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7
8      <!-- Bootstrap CSS -->
9      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
10
11     <title>Hello, world!</title>
12   </head>
13   <body>
14     <h1>Hello, world!</h1>
15
16     <!-- Optional JavaScript -->
17     <!-- jQuery first, then Popper.js, then Bootstrap JS -->
18     <script src="https://cdn.jsdelivr.net/npm/jquery/dist/jquery.min.js"></script>
19     <script type="text/javascript" src="objeto4.js"></script>
20   </body>
21 </html>
```

Quitamos esos JS y usamos
los JS nuestros

CRUD - Create, Read, Update and Delete

The image shows a web browser window with the address bar displaying `localhost/js/vue/practica5.html`. The page content is a large black text that says "Hello, world!". Below the page content, the browser's developer tools are open, showing the DOM tree, styles, and console. The DOM tree shows the `<body>` element containing an `<h1>Hello, world!</h1>` element. The styles panel shows the default browser styles for the `body` element, including `margin: 0;` and a font stack starting with `-apple-system`. The console is empty.

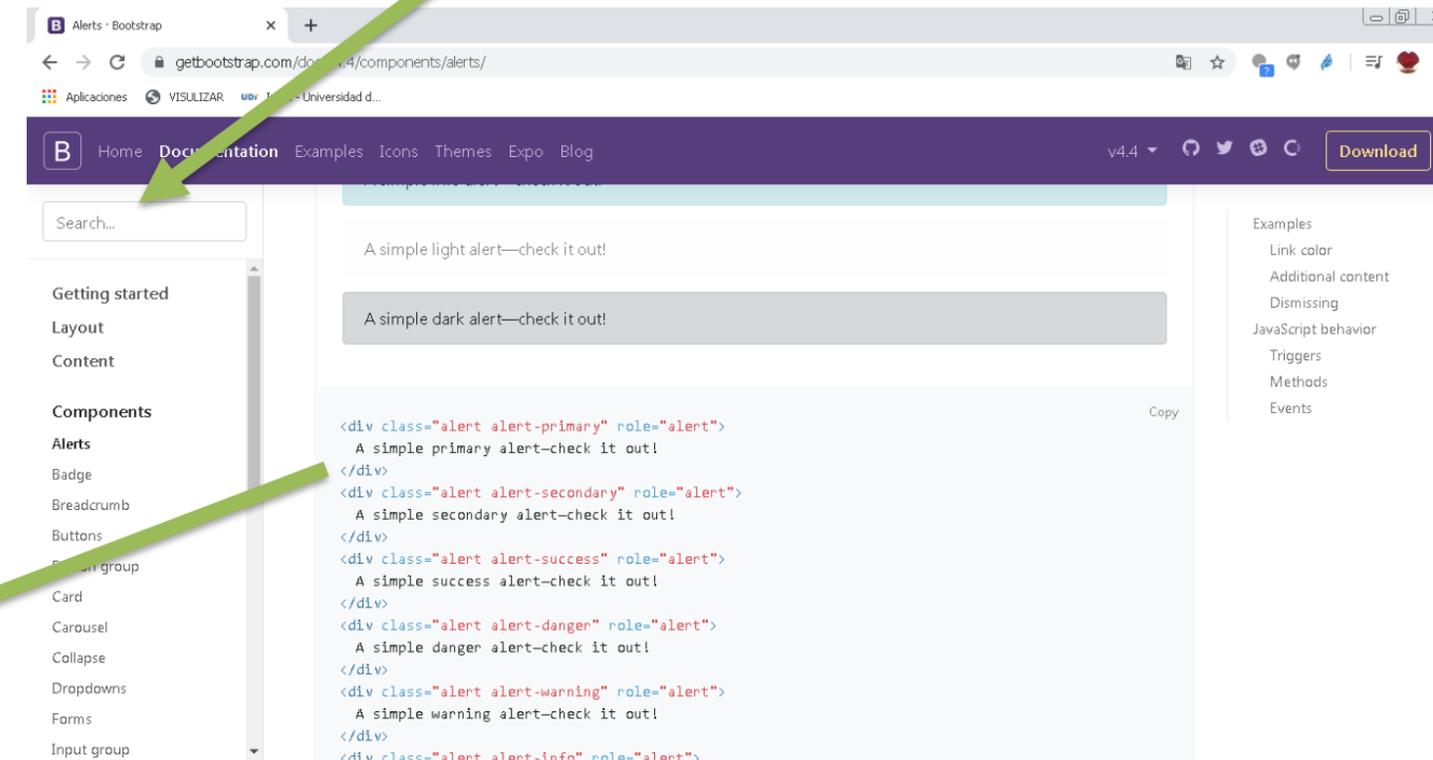
```
<!doctype html>
<html lang="en">
  <head>...</head>
  <body> == $0
    <h1>Hello, world!</h1>
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwjlyYfoR5Joz+n" crossorigin="anonymous"></script>
```

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto Sans", sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji";
}
```

CRUD - Create, Read, Update and Delete

Alert

```
<> practica5.html x JS objeto5.js
<> practica5.html > html > body > div#app.container.mt-5
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrin
7
8 <!-- Bootstrap CSS -->
9 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap
10
11 <title>Hello, World!</title>
12 </head>
13 <body>
14 <div class="container mt-5" id="app">
15 <h3>{{titulo}}</h3>
16 <input type="text" class="form-control my-3">
17 <button class="btn btn-primary">Ingresar</button>
18 <div class="mt-3">
19 <div class="alert alert-primary" role="alert">
20 A simple primary alert—check it out!
21 </div>
22 </div>
23 </div>
```



```
<> practica5.html JS objeto5.js x
JS objeto5.js > [?] app > [?] data > [?] titulo
1 const app = new Vue({
2   el: '#app',
3   data: {
4     titulo: 'CRUD con Vue'
5   },
6   methods: {
7
8   }
9 })
```

CRUD - Create, Read, Update and Delete

The screenshot displays a web browser at the URL `localhost/js/vue/practica5.html`. The page content includes a title "CRUD con Vue", an empty text input field, a blue "Ingresar" button, and a light blue alert box with the text "A simple primary alert—check it out!".

Overlaid on the right is the Vue DevTools component inspector. The component tree shows a `<div id="app" class="container mt-5">...</div>`. The code editor displays the following HTML and JavaScript:

```
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script type="text/javascript" src="objeto5.js"></script>
```

The Styles pane shows the default `body` styles from `_reboot.scss:47`:

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto Sans", sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji";
```

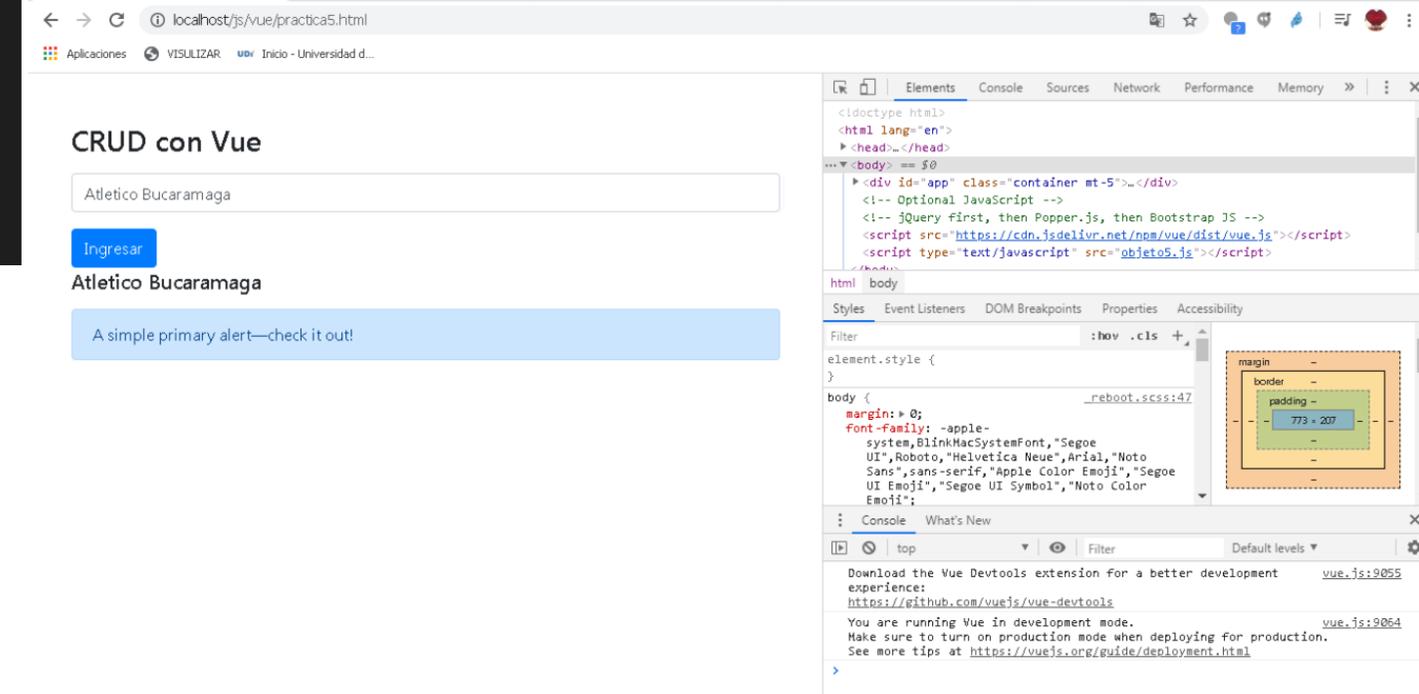
The DOM Breakpoints pane shows a box model diagram with dimensions `773 x 207`.

The Console pane displays the following messages:

- Download the Vue Devtools extension for a better development experience: <https://github.com/vuejs/vue-devtools> `vue.js:9055`
- You are running Vue in development mode. Make sure to turn on production mode when deploying for production. See more tips at <https://vuejs.org/guide/deployment.html> `vue.js:9064`

CRUD - Create, Read, Update and Delete

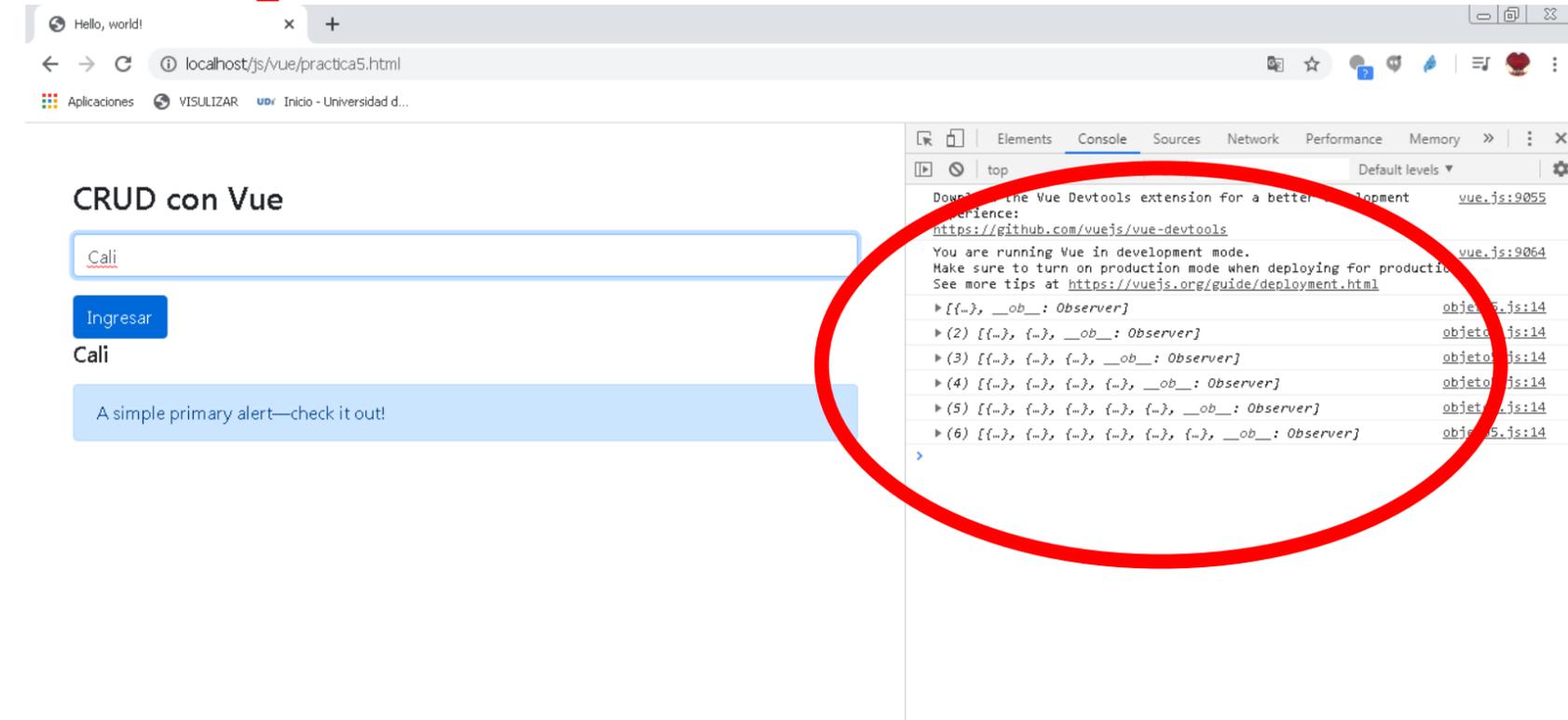
```
3 <div class="container mt-5" id="app">
4   <h3>{{titulo}}</h3>
5   <input type="text" class="form-control my-3" v-model="nuevaTarea">
6   <button class="btn btn-primary">Ingresar</button>
7   <h5>{{nuevaTarea}}</h5>
8   <div class="mt-3">
9     <div class="alert alert-primary" role="alert">
10      A simple primary alert—check it out!
11    </div>
12  </div>
13 </div>
```



```
<> practica5.html JS objeto5.js X
JS objeto5.js > [🔍] app > 🛠️ methods
1  const app = new Vue({
2  el: '#app',
3  data: {
4    titulo: 'CRUD con Vue',
5    tarea: [],
6    nuevaTarea: ''
7  },
8  methods: {
9
10 }
11 })
```

CRUD - Create, Read, Update and Delete

```
JS objeto5.js > [🔍] app > [🔑] methods > [📦] agregarTarea
1  const app = new Vue({
2  el: '#app',
3  data: {
4    titulo: 'CRUD con Vue',
5    tarea: [],
6    nuevaTarea: ''
7  },
8  methods: {
9    agregarTarea: function() {
10     this.tarea.push({
11       nombre: this.nuevaTarea,
12       estado: false
13     });
14     console.log(this.tarea);
15     this.nuevaTarea = '';
16   }
17 }
18 })
```



```
<div class="container mt-5" id="app">
  <h3>{{titulo}}</h3>
  <input type="text" class="form-control my-3" v-model="nuevaTarea"
  v-on:keyup.enter="agregarTarea">
  <button class="btn btn-primary" @click="agregarTarea">Ingresar</button>
  <h5>{{nuevaTarea}}</h5>
  <div class="mt-3">
    <div class="alert alert-primary" role="alert">
      A simple primary alert—check it out!
    </div>
  </div>
</div>
```

CRUD - Create, Read, Update and Delete

```
<> practica5.html x JS objeto5.js
container.mt-5 > div.mt-3 > div.alert.alert-primary > div.d-flex.justify-content-between.align-items-center >
11 </head>
12 <body>
13 <div class="container mt-5" id="app">
14 <h3>{{titulo}}</h3>
15 <input type="text" class="form-control my-3" v-model="nuevaTarea"
16 v-on:keyup.enter="agregarTarea">
17 <button class="btn btn-primary" @click="agregarTarea">Ingresar</button>
18 <div class="mt-3" v-for="(item, index) of tarea">
19 <div class="alert alert-primary" role="alert">
20 <div class="d-flex justify-content-between align-items-center">
21 <div>
22 | {{index}} - {{item.nombre}} - {{item.estado}}
23 </div>
24 <div>
25 <button class="btn btn-success btn-sm">OK</button>
26 <button class="btn btn-danger btn-sm">X</button>
27 </div>
28 </div>
29 </div>
30 </div>
31 </div>
```

```
<> practica5.html JS objeto5.js x
JS objeto5.js > [🔍] app > [🔧] methods > [📦] agregarTar
1 const app = new Vue({
2 el: '#app',
3 data: {
4 titulo: 'CRUD con Vue',
5 tarea: [],
6 nuevaTarea: ''
7 },
8 methods: {
9 agregarTarea: function(){
10 this.tarea.push({
11 nombre: this.nuevaTarea,
12 estado: false
13 });
14 console.log(this.tarea);
15 this.nuevaTarea='';
16 }
17 }
18 })
```

CRUD - Create, Read, Update and Delete

The screenshot displays a web browser window with a single tab titled "Hello, world!". The address bar shows the URL "localhost/js/vue/practica5.html". The browser's taskbar at the bottom includes icons for "Inicio", "e", "File Explorer", "Google Chrome", "Microsoft Teams", "PowerPoint", and "Visual Studio Code".

The main content of the browser is a web application titled "CRUD con Vue". It features a search input field containing the text "Millonarios" and a blue "Ingresar" button. Below the input, there is a list of four items, each in a light blue box with a green "OK" button and a red "X" button:

- 0 - Bucaramanga - false
- 1 - Nacional - false
- 2 - America - false
- 3 - Junior - false

The right side of the browser window shows the developer console with the "Console" tab selected. The console displays several messages:

- A message from "vue.js:9055" suggesting to download the Vue Devtools extension: "Download the Vue Devtools extension for a better development experience: https://github.com/vuejs/vue-devtools".
- A message from "vue.js:9064" stating: "You are running Vue in development mode. Make sure to turn on production mode when deploying for production. See more tips at https://vuejs.org/guide/deployment.html".
- Four log entries from "objeto5.js:14" showing object structures:
 - ▶ [{"__ob__": Observer}]
 - ▶ (2) [{"__ob__": Observer}, {"__ob__": Observer}]
 - ▶ (3) [{"__ob__": Observer}, {"__ob__": Observer}, {"__ob__": Observer}]
 - ▶ (4) [{"__ob__": Observer}, {"__ob__": Observer}, {"__ob__": Observer}, {"__ob__": Observer}]

The bottom of the console shows the "Console" and "What's New" tabs, and the system tray at the bottom right indicates the time "14:13" and date "25/04/2020".

CRUD - Create, Read, Update and Delete

```
<> practica5.html X JS objeto5.js
<> practica5.html > html > body > div#app.container.mt-5 > div.mt-3 > div.alert.alert-prim
13 <div class="container mt-5" id="app">
14   <h3>{{titulo}}</h3>
15   <input type="text" class="form-control my-3" v-model="nuevaTarea"
16   v-on:keyup.enter="agregarTarea">
17   <button class="btn btn-primary" @click="agregarTarea">Ingresar</button>
18   <div class="mt-3" v-for="(item, index) of tarea">
19     <div class="alert alert-primary" role="alert">
20       <div class="d-flex justify-content-between align-items-center">
21         <div>
22           {{index}} - {{item.nombre}} - {{item.estado}}
23         </div>
24         <div>
25           <button class="btn btn-success btn-sm" @click="editarTarea(index)"
26           ">OK</button>
27           <button class="btn btn-danger btn-sm">X</button>
28         </div>
29       </div>
30     </div>
31 </div>
```

```
<> practica5.html JS objeto5.js X
JS objeto5.js > app > methods > agregarTarea
1  const app = new Vue({
2  el: '#app',
3  data: {
4    titulo: 'CRUD con Vue',
5    tarea: [],
6    nuevaTarea: ''
7  },
8  methods: {
9    agregarTarea: function(){
10     this.tarea.push({
11       nombre: this.nuevaTarea,
12       estado: false
13     });
14     this.nuevaTarea='';
15   },
16   editarTarea: function(index){
17     console.log('editar',index);
18   }
19 }
20 })
```

CRUD - Create, Read, Update and Delete

The screenshot shows a web browser window with the address bar at `localhost/js/vue/practica5.html`. The page title is "Hello, world!". The main content area displays a CRUD application titled "CRUD con Vue". It features a search input field, an "Ingresar" button, and a list of four items, each with "OK" and "X" buttons:

- 0 - Bucaramanga - false
- 1 - Nacional - false
- 2 - America - false
- 3 - Medellin - false

The browser's developer console is open, showing the following messages:

```
Download the Vue Devtools extension for a better development experience:
https://github.com/vuejs/vue-devtools

You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html

editar 1 objeto5.js:17
editar 2 objeto5.js:17
editar 0 objeto5.js:17
editar 3 objeto5.js:17
```