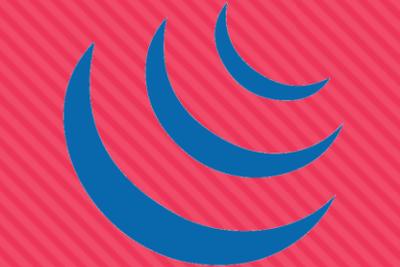


Programación Web – 6 JQuery

Por: Ing. Luis Daniel Lepe Rodríguez

The jQuery logo consists of three blue, curved, overlapping shapes that resemble a stylized 'J' or a series of arcs, positioned above the text 'jQuery'.

JQuery

jQuery

JQuery es una librería de JavaScript que permite ahorrar código y simplificar operaciones.

Esta librería de código abierto, simplifica la tarea de programar en JavaScript y **permite agregar interactividad a un sitio web** sin tener conocimientos del lenguaje. La manipulación del DOM, los manejos de eventos, animaciones y AJAX son muchísimo más sencillos de realizar utilizando esta herramienta.

¿Por qué es JQuery tan bueno?

Cosas que hace bien:

- Simplifica AJAX
- Manipulación del DOM
- Administración de eventos
- Animaciones
- Normaliza diferencias entre navegadores

¿Por qué se usa tanto?

- Está muy bien diseñado
- Es muy fácil de extender con plugins
- Muy buena documentación
- Una comunidad muy grande
- Compatible entre navegadores



¿Cómo usar JQuery?

Incluir JQuery

```
<body>
  <div>
    <a id="toggleContent" href="#">Activar cambio</a>
    <div id="content">

  </div>
</div>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"
  integrity="sha256-/xUj+30JU5yEx1q6GSYGS7k7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>
<script>
  $(document).ready(function () {
    $('#toggleContent').bind('click', function () {
      $('#content').text('Hola mundo');
      return false;
    });
  });
</script>
</body>
```

Nuestro DOM

Código que usa JQuery

Filosofía de JQuery

Encontrar un elemento

```
<script>  
    $('#container').addClass('closed');  
</script>
```

Hacer algo con el

La función del dólar \$()

- Los identificadores de JQuery suelen empezar con `$()`
- El framework de JQuery automáticamente asigna '\$' a la función 'jQuery'

```
$ === jQuery //true
```

```
$(selector) es lo mismo que jQuery(selector)
```

- Se puede desasignar que JQuery use '\$' por defecto

```
$.noConflict();
```

```
$ === jQuery //false
```

- Se puede reasignar JQuery a otra variable

```
$j = $.noConflict();
```

```
$j === jQuery //true
```

```
$j === $ //false
```

```
$ === jQuery //false
```

Document Ready Event

- La función `$(document).ready()` ofrece una manera de ejecutar código JavaScript tan pronto como el DOM de nuestra página sea seguro de manipular. Este suele ser un buen momento para realizar las tareas necesarias antes de que el usuario vea o interactúe con la página, por ejemplo, para agregar controladores de eventos e inicializar complementos.

```
$(document).ready(function () {  
    // El DOM está listo  
    $('#toggleContent').bind('click', function () {  
        $('#content').text('Hola mundo');  
        return false;  
    })  
});
```

JQuery Selectors

JQuery	Selecciona...
<code>\$('#id');</code>	Por ID
<code>\$('.class');</code>	Por clase
<code>\$('#div');</code>	Por Elemento
<code>\$('#div, span, #id, .class');</code>	Múltiples tipos
<code> \$('*');</code>	Todo
<code> \$('#div p');</code>	Descendientes
<code> \$('#input[type=text]');</code>	Todos los elementos con un atributo específico

Funciones esenciales

- Tenemos 'text', 'val', 'attr', 'html', como Getters (obtener) y Setters (establecer):
 - Getter (regresa un string)

```
var texto = $('#id').text();
```
 - Setter (Modifica el DOM)

```
$('#id').text('Hola mundo');
```
- text() obtiene el contenido de texto estático.
- val() se utiliza para sacar información de inputs.
- attr() sirve para obtener los atributos, puede tomar JSON.
- html() se utiliza para el contenido incluyendo la etiqueta HTML.

Atributos y clases

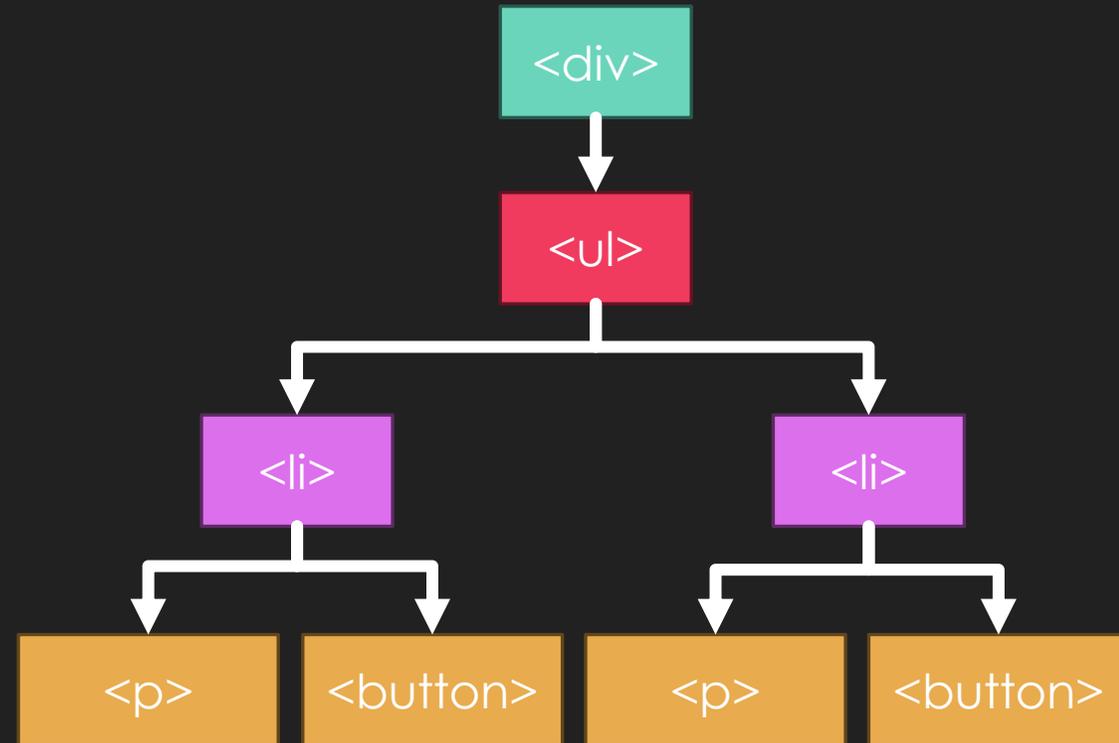
Agregar o remover para clases y atributos

- `$('#id').attr('atributoNombre');` // Obtener el valor de un atributo
- `$('#id').attr('atributoNombre', 'valor');` // Establecer el valor de un atributo
- `$('#id').removeAttr('atributoNombre');` // Eliminar un atributo de la etiqueta

- `$('#id').addClass('claseNombre');`
- `$('#id').removeClass('claseNombre');`
- `$('#id').toggleClass('claseNombre');`

Navegar por el DOM (traversing)

- JQuery traversing, que “moverse”, se utiliza para buscar o navegar entre elementos HTML en función a su relación con otros elementos.
- Puede moverse fácilmente hacia arriba (antepasados), hacia abajo (descendientes) y hacia los lados (hermanos) en el árbol, comenzando desde el elemento seleccionado (actual). Este movimiento se denomina “traversing” (o moverse a través) del árbol DOM.



Navegar por el DOM (traversing)

Compra de comida online

Productos en el carrito

Nombre Producto	Cantidad	Precio Unitario	Precio Total
Papas fritas	- 1 +	US\$ 3.50	US\$ 3.50
Hamburguesa	- 1 +	US\$ 10.00	US\$ 10.00
Pizza	- 1 +	US\$ 15.50	US\$ 15.50
Completo	- 1 +	US\$ 8.50	US\$ 8.50
Vaciar carrito		Precio Total	US\$ 37.50



Papas fritas

Papas frescas preparadas en el momento, caseras, seleccionadas exclusivamente para tener una buena calidad y con un sabor inigualable.

US\$ 3.50

[Agregar](#)



Hamburguesa

La mejor Hamburguesa del mercado, con productos finamente seleccionados, una presentación única y el mejor sabor.

US\$ 10.00

[Agregar](#)



Pizza

Masa preparada en el local, fermentada el tiempo suficiente para que de una mordida puedas sentir una sensación única de sabor y textura.

US\$ 15.50

[Agregar](#)



Completo

Estos son los mejores completos de la ciudad, hechos con el mejor pan y ingredientes finamente seleccionados.

US\$ 8.50

[Agregar](#)

Navegar por el DOM (traversing)

Familia

- `.parent()`
- `.siblings()`
- `.children()`

Proximidad

- `.closest()`
- `.next()`
- `.prev()`
- `.nextAll()`
- `.prevAll()`

Búsqueda

- `.find()`

Manejo de eventos

- `$('#id').on('handler', function(){}); //Para todos los eventos`

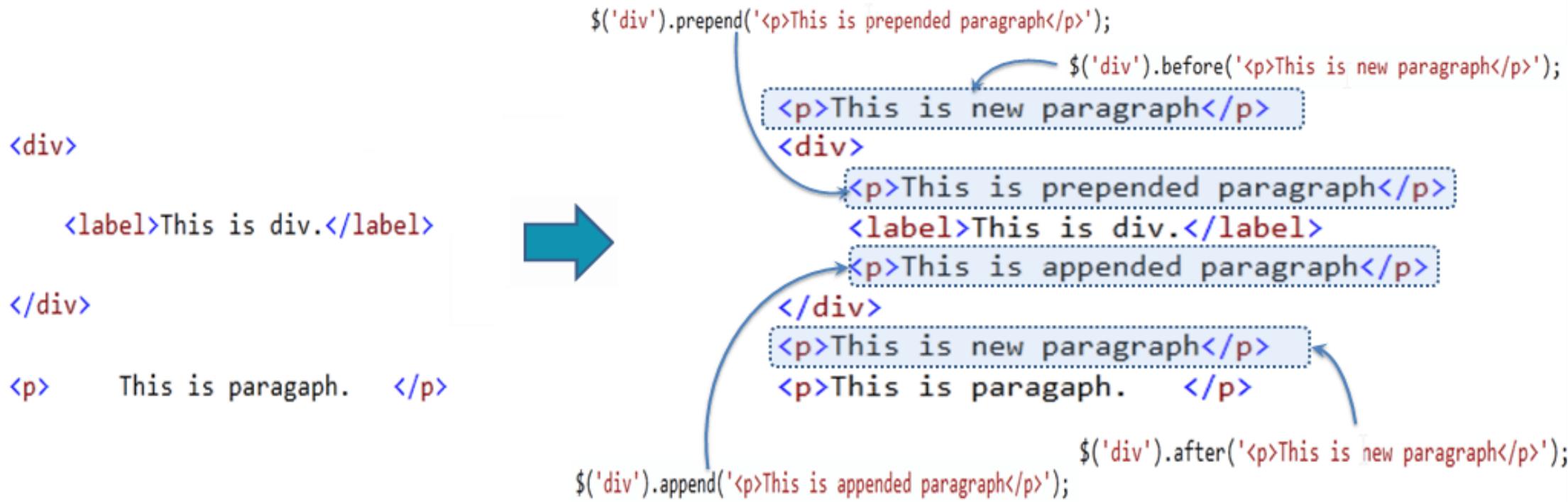
- Ejemplo...

```
$('.btnEliminar').on('click', function(){  
    // Funcionalidad a realizar  
    $(this).parent().remove();  
});
```

Manipulación de DOM

Método	Descripción
<code>.prepend()</code>	Inserta contenido como hijo al inicio del selector especificado.
<code>.append()</code>	Inserta el contenido como hijo al final del selector especificado.
<code>.before()</code>	Inserta contenido como hermano al inicio del selector especificado.
<code>.after()</code>	Inserta contenido como hermano al final del selector especificado.
<code>.remove()</code>	Elimina el elemento(s) del DOM que especifica el selector.
<code>.replaceAll()</code>	Reemplaza el elemento(s) especificado con el elemento especificado.
<code>.wrap()</code>	Envuelve una estructura HTML alrededor de cada elemento especificado por el selector.

Manipulación de DOM



AJAX

AJAX = **A**synchronous **J**avaScript **A**nd **X**ML.

AJAX no es un lenguaje de programación.

AJAX solo usa una combinación de:

- Un objeto XMLHttpRequest integrado en el navegador (para solicitar datos de un servidor web)
- JavaScript y HTML DOM (para mostrar o utilizar los datos)



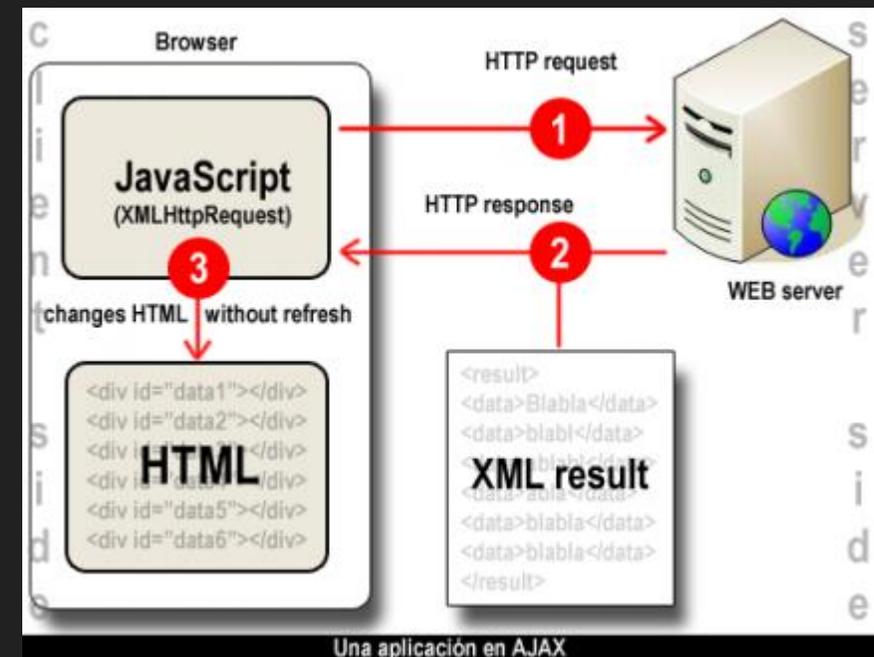
AJAX es un nombre engañoso. Las aplicaciones AJAX pueden usar XML para transportar datos, pero actualmente es mucho más común transportar datos como texto sin formato o texto JSON.

[What is AJAX - W3Schools](#)

AJAX

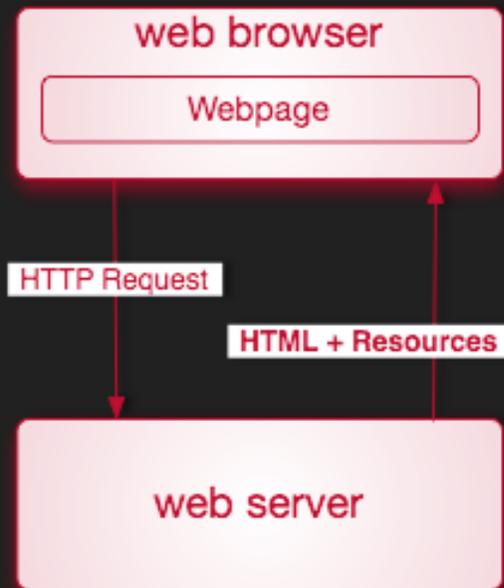
AJAX permite que las páginas web se actualicen de forma asincrónica mediante el intercambio de datos con un servidor web detrás de escena. Esto significa que es posible actualizar partes de una página web, sin recargar toda la página.

1. Se produce un evento en una página web (se carga la página, se hace clic en un botón)
2. JavaScript crea un objeto XMLHttpRequest
3. El objeto XMLHttpRequest envía una solicitud a un servidor web.
4. El servidor procesa la solicitud.
5. El servidor envía una respuesta a la página web.
6. La respuesta es leída por JavaScript.
7. JavaScript realiza la acción adecuada (como la actualización de la página)

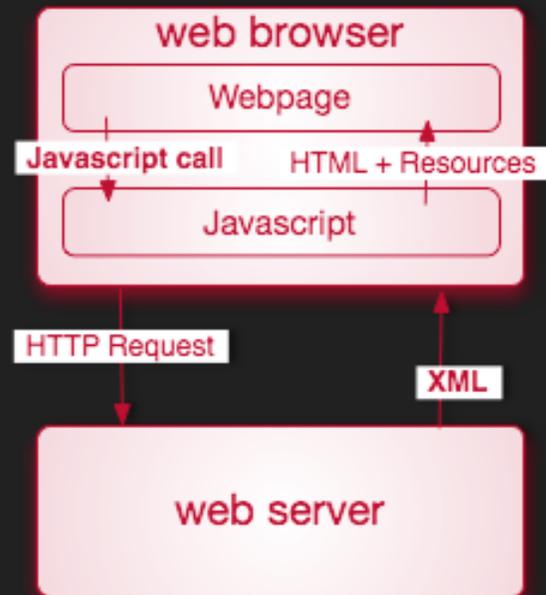


AJAX

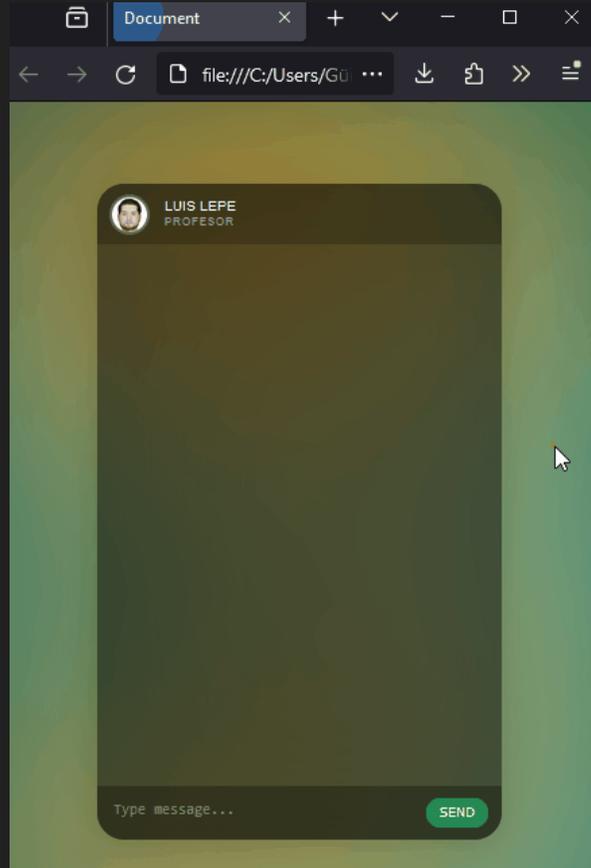
Traditional web model



AJAX web model

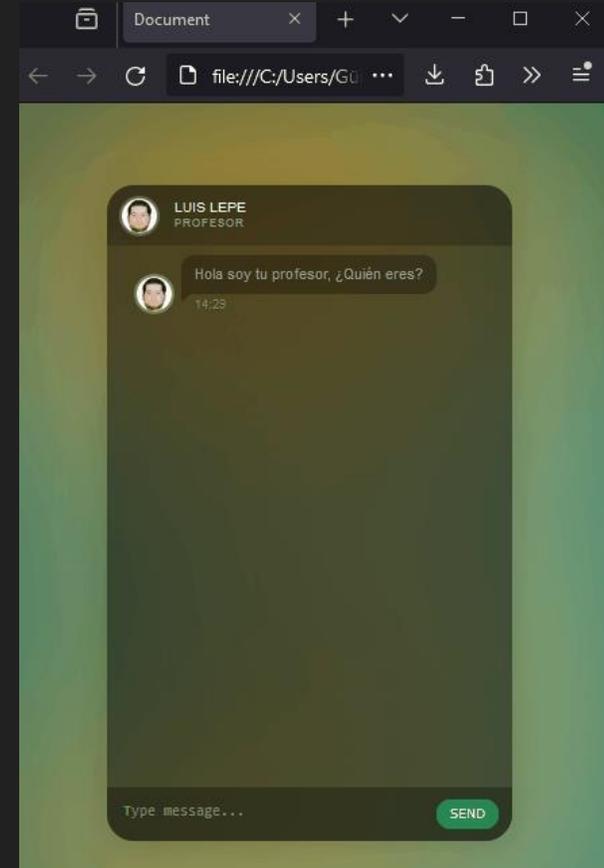


Con AJAX



Actualización de contenido parcial, sin necesidad de refrescar toda la página

Sin AJAX



Se debe actualizar toda la página para recibir nuevo contenido.

Peticiones HTTP o verbos HTTP

El protocolo HTTP define una serie de métodos de solicitud para indicar la acción que se desea realizar para un recurso determinado.

Veremos solo 4 peticiones que son las que están relacionadas con los [API RESTful](#):

- GET
- PUT
- POST
- DELETE



Códigos de respuesta HTTP

Los códigos de estado de respuesta HTTP indican si una solicitud HTTP específica se ha completado correctamente. Las respuestas se agrupan en cinco clases:

1. Respuestas informativas (**100-199**)
2. Respuestas satisfactorias (**200-299**)
3. Redirecciones (**300-399**)
4. Errores del cliente (**400-499**)
5. Errores del servidor (**500-599**)



DOM

