

CODES

Capacitación Web – AJAX

ÍNDICE

1. AJAX

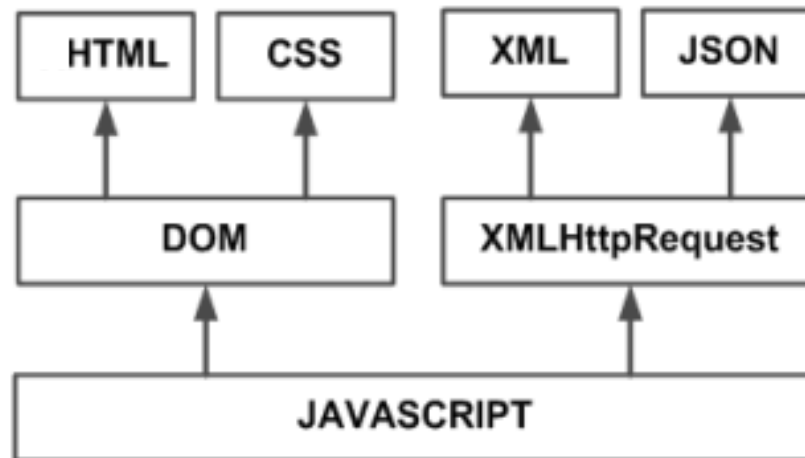
CONCEPTOS

- **Ajax**: -acrónimo de **Asynchronous JavaScript And XML** (Extensible Markup Language)
 - es una técnica de desarrollo web que permite la comunicación entre el cliente y el servidor de manera asincrónica, **no es un lenguaje de programación**, sino que utiliza conjuntamente varias tecnologías existentes para crear aplicaciones web.
 - Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios, mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. Es posible lograr aplicaciones web capaces de actualizarse continuamente sin tener que volver a cargar la página completa. mejorando la interactividad, velocidad y usabilidad.
- **HttpHandler**: Un ASP.NET HTTP handler es el proceso que se ejecuta en respuesta a una solicitud realizada a una aplicación web ASP.NET. El manejador (handler) más común en ASP.NET es el que procesa los archivos .aspx (las páginas) y también el .ashx que es el que permite procesar peticiones.

1. AJAX

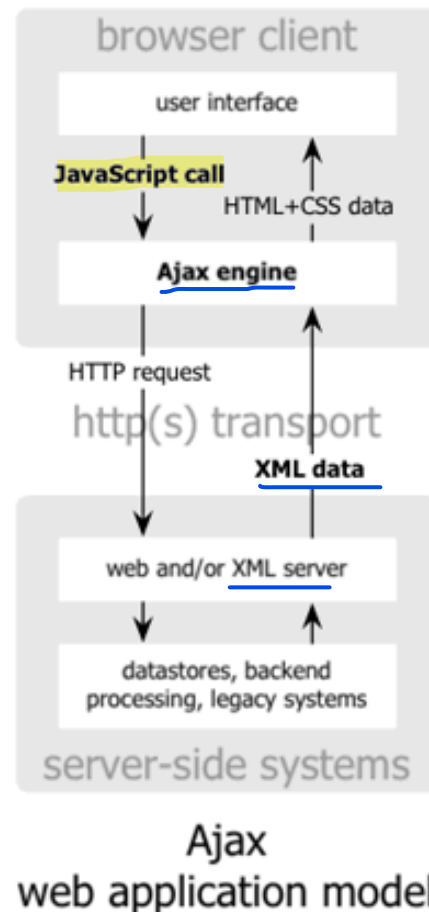
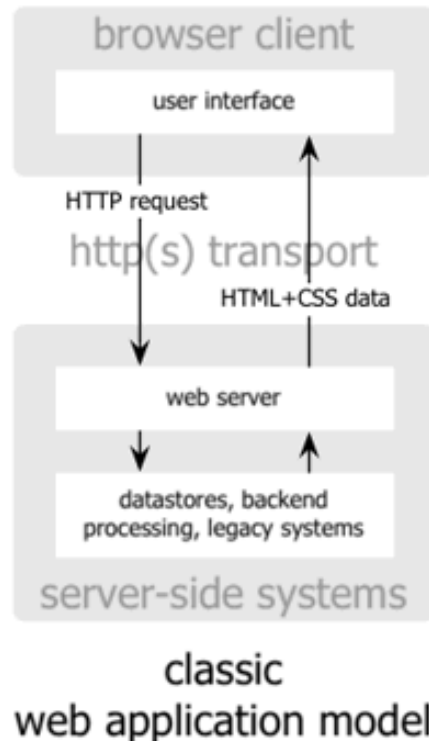
Las tecnologías que forman AJAX son:

- HTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.



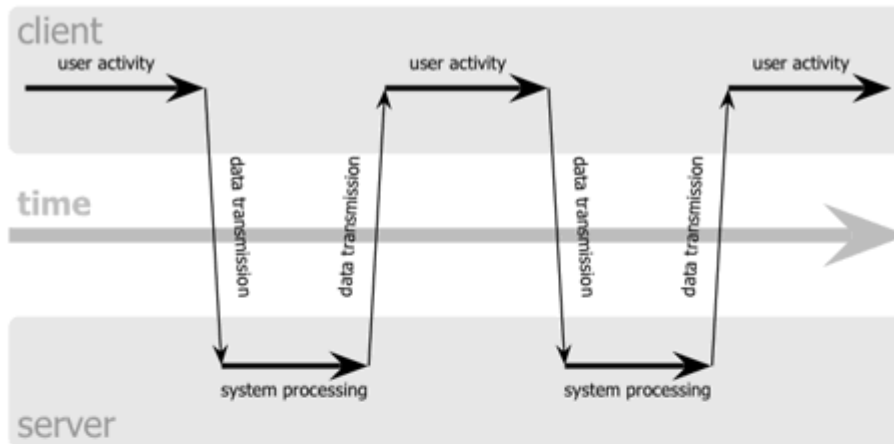
1. AJAX

En el siguiente esquema, la imagen de la izquierda muestra el modelo tradicional de las aplicaciones web. La imagen de la derecha muestra el nuevo modelo propuesto por AJAX:



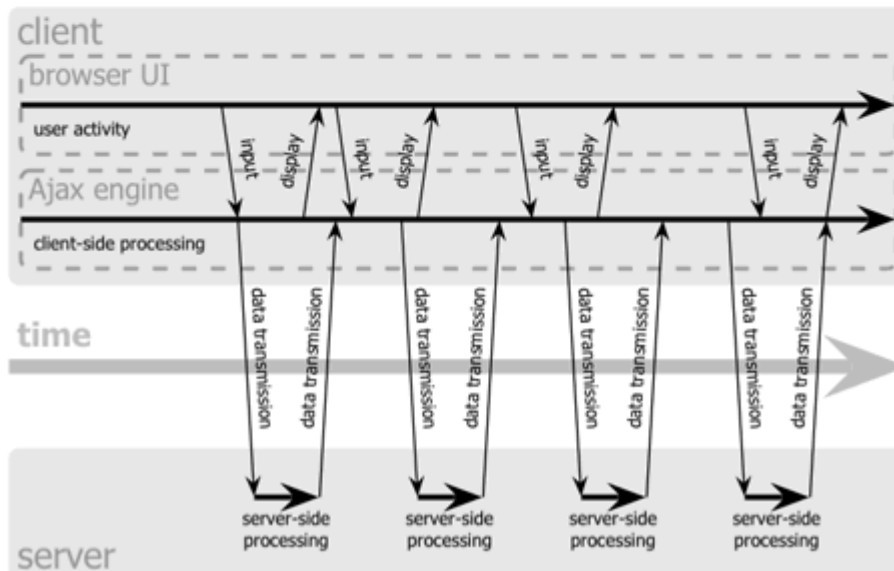
1. AJAX

classic web application model (synchronous)



El esquema de la izquierda muestra la diferencia más importante entre una aplicación web tradicional y una aplicación web creada con AJAX. La imagen superior muestra la interacción síncrona propia de las aplicaciones web tradicionales. La imagen inferior muestra la comunicación asíncrona de las aplicaciones creadas con AJAX.

Ajax web application model (asynchronous)



Antes de entrar en un ejemplo práctico de Ajax, introduciremos el concepto de JSON:

JSON (JavaScript Object Notation) es la forma de denotar objetos en javascript. Es un formato sencillo que define una estructura de datos para el intercambio de información- La notación de objetos mediante JSON es una de las características principales de JavaScript y es un mecanismo definido en los fundamentos básicos del lenguaje.

En los últimos años, JSON se ha convertido en una alternativa al formato XML, ya que es más fácil de leer y escribir, además de ser mucho más conciso. No obstante, XML es superior técnicamente porque es un lenguaje de marcado, mientras que JSON es simplemente un formato para intercambiar datos. ?

La especificación completa de JSON se puede consultar en RFC 4627 y su tipo MIME oficial es application/json.

EJEMPLO JSON:

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        {
          "value": "New",
          "onclick": "CreateNewDoc"
        },
        {
          "value": "Open",
          "onclick": "OpenDoc"
        },
        {
          "value": "Close",
          "onclick": "CloseDoc"
        }
      ]
    }
  }
}
```


Ejemplo Ajax:

Vamos a usar, a modo de servidor, para hacer la petición, una API de chistes random.

URL: <https://official-joke-api.appspot.com/jokes/programming/random>

Y vamos a ver, en la práctica, como se realiza con el método tradicional y también utilizando AJAX para ver alguna de las diferencias de las que charlamos.

- Modo sin AJAX:

```
1 <html>
2   <head>
3     <title>Ejemplo Sin AJAX</title>
4   </head>
5   <body>
6     <form type="GET" action="https://official-joke-api.appspot.com/jokes/programming/random">
7       <input type="submit" value="Contar Chiste"></input>
8     </form>
9   </body>
10 </html>
```

- Modo con AJAX:

```
<html>
  <head>
    <script type="text/javascript" src="main.js"></script>
    <title>Ejemplo Con AJAX</title>
  </head>
  <body>
    <input type="button" value="Contar Chiste" onclick="onClickEnviar()"></input><br/>
    <label>Setup:</label>
    <input type="text" value="" style='width:800px' id="txtSetup"></input><br/>
    <label>Punchline:</label>
    <input type="text" value="" style='width:800px' id="txtPunchline"></input>
  </body>
</html>
```

- Modo con AJAX: .js

```
1  function onClickEnviar(){
2      //Objeto para realizar la peticion AJAX
3      var peticion = new XMLHttpRequest();
4
5      //Indico que se va a ejecutar una vez que la misma finalice
6      peticion.onreadystatechange = function(){
7          if(this.readyState == 4 && this.status==200){
8              handlerResultado(this);
9          }
10     }
11     //Indico a donde voy a hacer la peticion
12     peticion.open("GET", "https://official-joke-api.appspot.com/jokes/programming/random");
13
14     //Disparo la peticion
15     peticion.send();
16 }
17
18 function handlerResultado (This) {
19
20     var res = JSON.parse(This.responseText)[0];
21     document.getElementById("txtSetup").value = res.setup;
22     document.getElementById("txtPunchline").value = res.punchline;
23 }
```

- Modo con AJAX y JQuery:

```
<html>
  <head> |
    <script type="text/javascript" src="https://code.jquery.com/jquery-3.6.0.js"></script>
    <script type="text/javascript" src="mainJQ.js"></script>
    <title>Ejemplo Con AJAX y JQuery</title>
  </head>
  <body>
    <input type="button" id="ajaxBtn" value="Contar Chiste" onclick="onClickEnviar()"></input><br/>
    <label>Setup:</label>
    <input type="text" value="" style='width:800px' id="txtSetup"></input><br/>
    <label>Punchline:</label>
    <input type="text" value="" style='width:800px' id="txtPunchline"></input>
  </body>
</html>
```

- Modo con AJAX y JQuery: .js

```
1  function onClickEnviar(){
2      $.ajax({
3          method: "GET",
4          url: "https://official-joke-api.appspot.com/jokes/programming/random",
5          dataType: 'json',
6          success: function (data){
7              handlerResultado(data);
8          },
9          error: function (jqXHR, textStatus, errorMessage) { // error callback
10             console.log('Error: ' + errorMessage);
11         }
12     });
13 }
14
15 function handlerResultado (data) {
16     var res = data[0];
17     $("#txtSetup").val(res.setup);
18     $("#txtPunchline").val(res.punchline);
19 }
20
```

FIN

Gracias