

Ingeniería de Sistemas

Desarrollo y Servicios Web

Sesión 8

Fernando Barraza A.
fbarraza@javerianacali.edu.co

Sesión 8



- Objetivo: Brindar al estudiante los conocimientos teóricos y prácticos alrededor de AJAX
- Temas:
 - Que es AJAX?
 - Componentes de AJAX
 - Principios básicos de AJAX
 - Arquitectura de una aplicación AJAX
 - Ejemplo de una aplicación AJAX

Que es AJAX?



- Es una implementación de una técnica para la ejecución remota de programas de scripting
- Elimina complejidad de invocación de objetos remotos (Ej.: CORBA)
- Hace uso intensivo de estándares XML
- Su antecesor fue XML-RPC

Significado del acrónimo AJAX



- **XHTML** (o HTML) y hojas de estilos en cascada (**CSS**) para el diseño que acompaña los datos
- **Document Object Model** (DOM) accedido con un lenguaje de scripting por parte del usuario
 - Especialmente implementaciones ECMAScript como **JavaScript** y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto **XMLHttpRequest** para intercambiar datos asincrónicamente con el servidor web.
 - En algunos frameworks y en algunas situaciones concretas, se usa un objeto **iframe** en lugar del **XMLHttpRequest** para realizar dichos intercambios.
- **XML** para la transferencia de vuelta al servidor
 - Cualquier formato puede funcionar, incluyendo HTML preformateado como texto plano o JSON.

Cambio de Paradigma con AJAX



Web-Based Application Model



"Dumb Client" simply renders HTML in the browser

HTML (UI content)



Server provides business logic and generates UI markup

Browser-Based Application Model



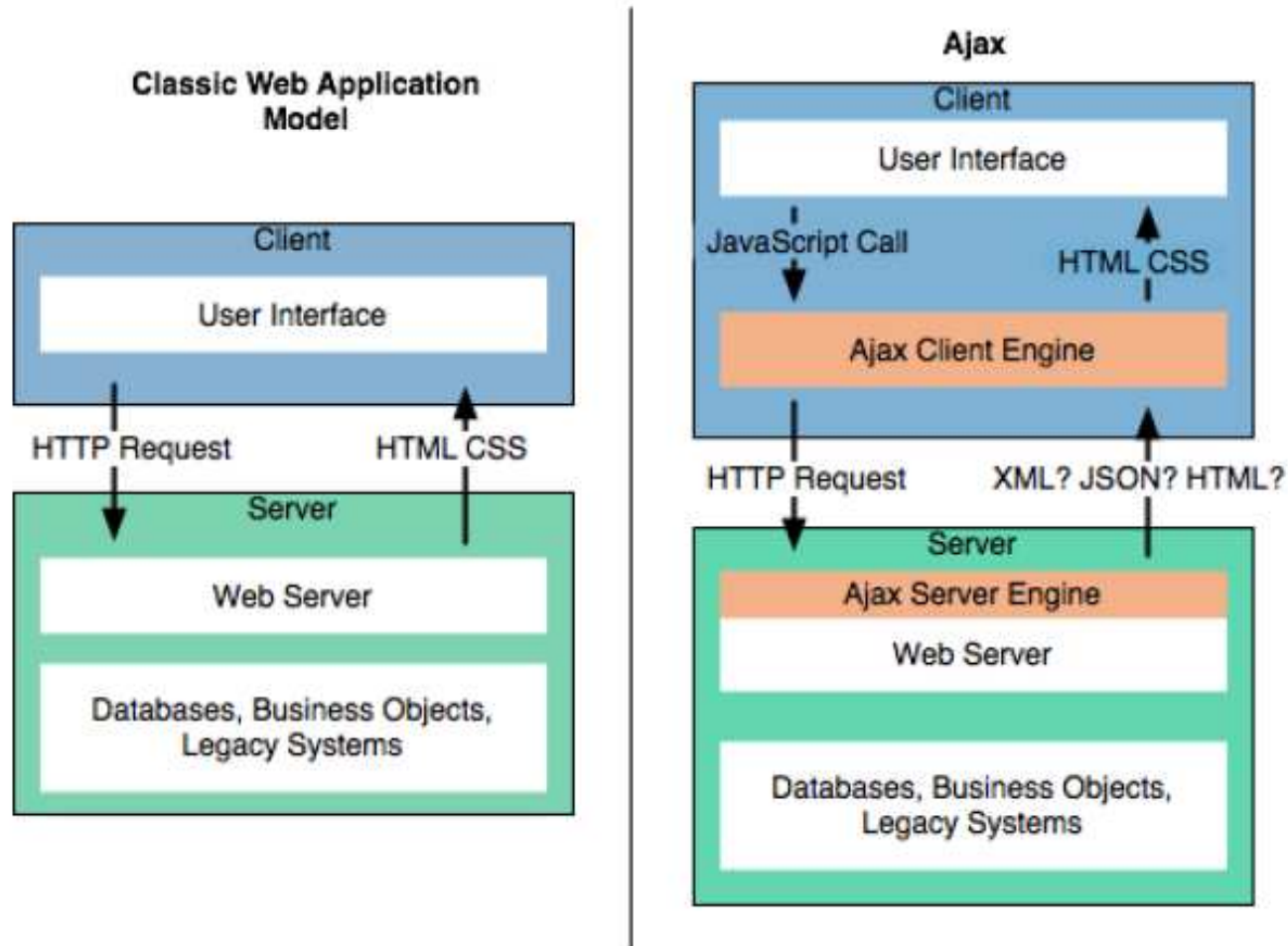
Application UI logic runs in browser.

Data (e.g. via SOAP)



Server exports application API. Performs core business logic

Arquitectura de AJAX vs. Aplicación web tradicional



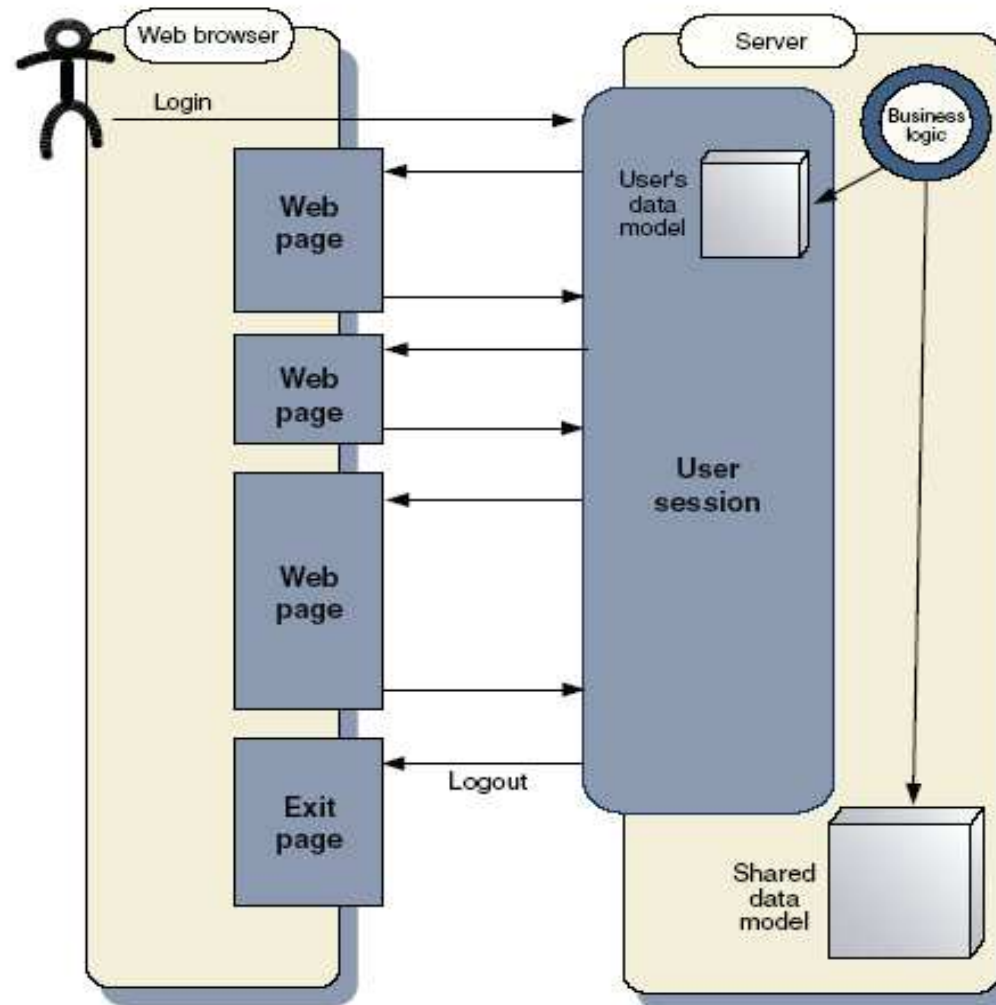
4 principios para AJAX



1. *El browser hospeda parte de la aplicación:*

- En una página Web tradicional, el browser es una terminal “tonta” porque no conoce nada acerca de cómo interactúa el usuario con la aplicación.
- Toda la información es mantenida en el servidor Web, en la sesión del usuario

Aplicación Web Tradicional

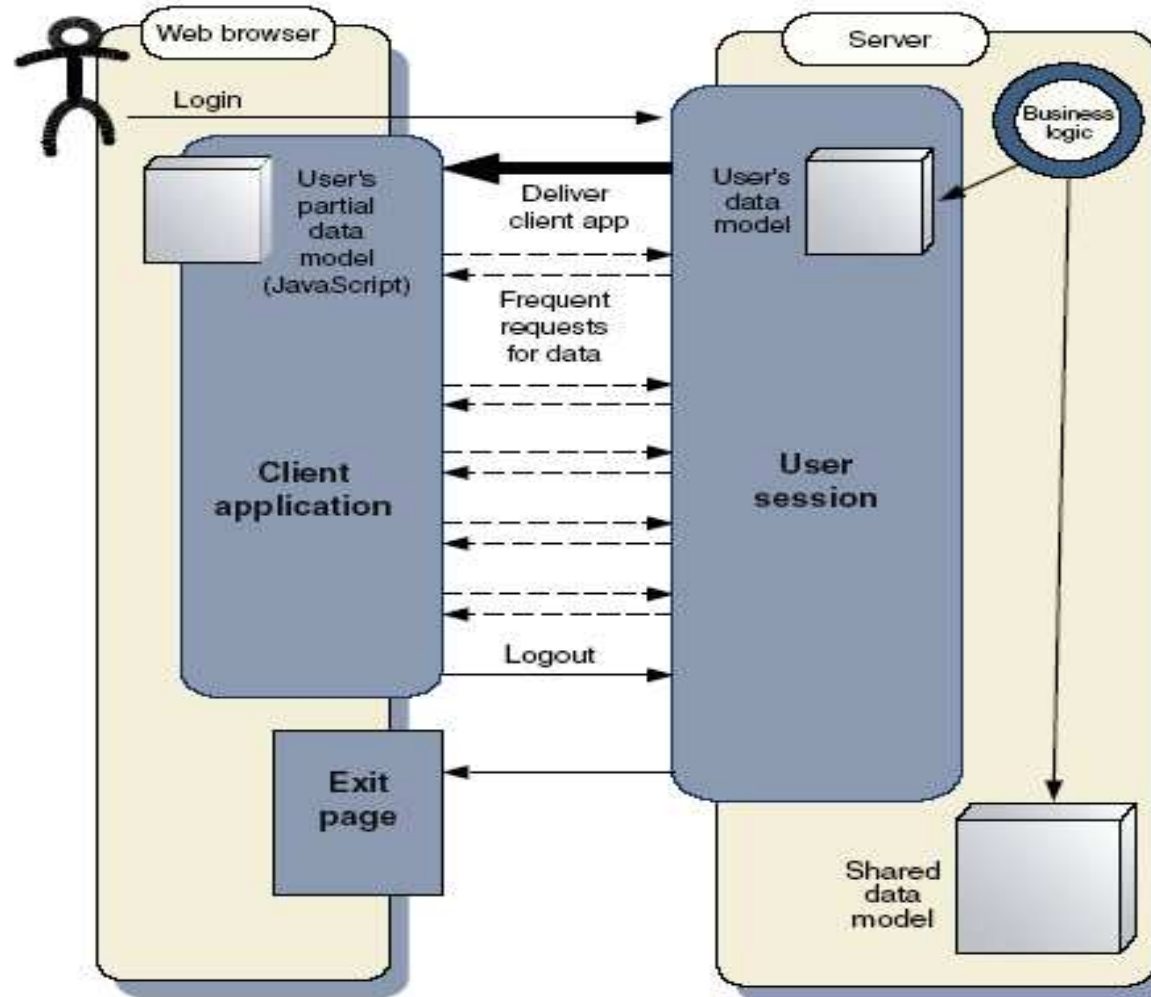


El browser hospeda la aplicación, no la contiene (cont)



- Una aplicación AJAX mueve algo de la lógica hacia el browser
- Cuando un usuario se logea, un documento más complejo que simple HTML es enviado al navegador, del cual una gran parte es código en JavaScript
- El documento permanece en la sesión del usuario y probablemente cambiará mientras el usuario interactúa con él
- La lógica embebida en el documento sabe como responder a las entradas del usuario y decidir si manejar dichas entradas o pasar el requerimiento al servidor Web (quien tiene acceso a la base de datos por ejemplo) o una combinación de ambos.

Aplicación AJAX



Principios (cont)



2. *El servidor entrega datos, no contenido:*

- En una aplicación Web tradicional el servidor Web entrega una mezcla de datos y contenido en cada paso
- Por ejemplo en un carrito de compras, si un usuario adiciona un ítem, lo que se debe hacer es actualizar es la cuenta total. No reenviar el contenido de todos los ítems actuales en el carrito, ni las listas de navegación, estilos de presentación, etc.
- Una aplicación AJAX en cambio retorna un fragmento de JavaScript, o un pequeño documento XML.

Principios (cont)



3. *La interacción con el usuario puede ser continua y fluida:*

- En una aplicación Web tradicional mientras la página es enviada, el usuario está en el limbo. Aunque la página vieja está aún visible, el browser le permite al usuario hacer click en los enlaces visibles con resultado impredecibles.
- La aplicación AJAX envía los datos asincrónicamente, lo que permite al usuario hacer varios clicks rápidamente (como por ejemplo cargar varios ítems en un carrito de compras simultáneamente) y saber que está pasando realmente.

Principios (cont)



- *Requiere disciplina para codificación (patrones)*
 - Una aplicación AJAX es una pieza de código compleja que se comunica eficientemente con el servidor mientras el usuario trabaja.
 - Aunque tiene claras diferencias con una aplicación tradicional (basada en páginas Web) las diferencias no son tan dramáticas por lo que requiere conocer muy bien donde hacer énfasis en las fortalezas de usar los principios anteriores.

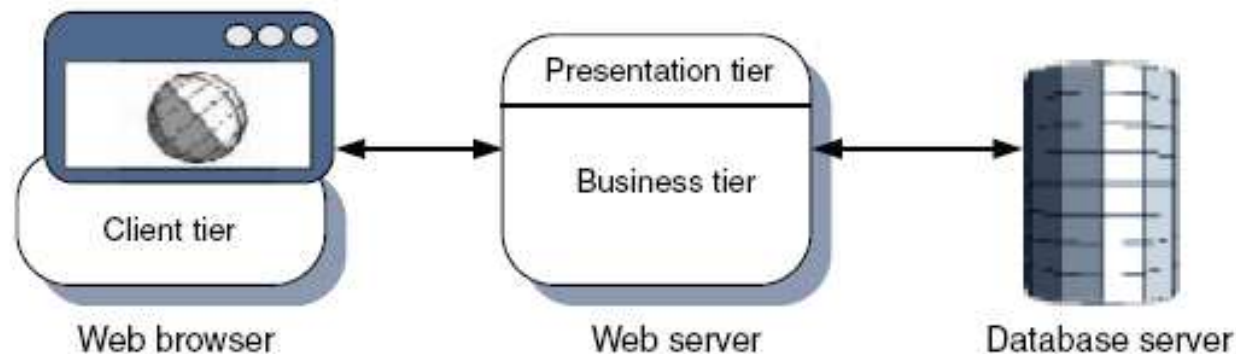
Consideraciones sobre AJAX



- Ventajas
 - Portabilidad, soportada entre capas de la arquitectura de la aplicación
 - RIA (Rich-Internet-Applications), aplicaciones Web más ricas en la interacción con el usuario
 - Manejo de datos estructurado
 - Mejor desempeño (no para todos los casos)
- A tener en cuenta
 - Arquitectura de la aplicación
 - Lógica en Javascript
 - El framework a utilizar

Consideraciones (1)

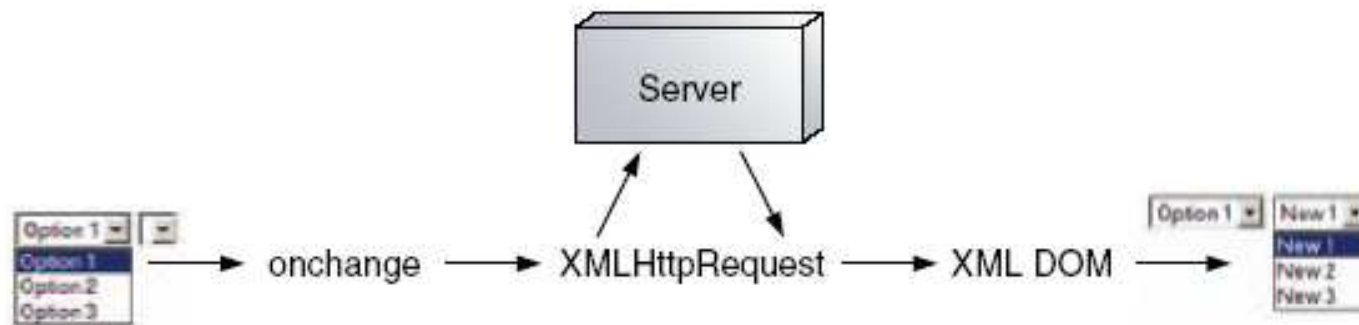
Portabilidad: Una aplicación AJAX permite reasignar algunas de las responsabilidades de la capa de presentación desde el servidor hacia el navegador, en una nueva entidad llamada ahora la capa cliente de la aplicación



Consideraciones (2)



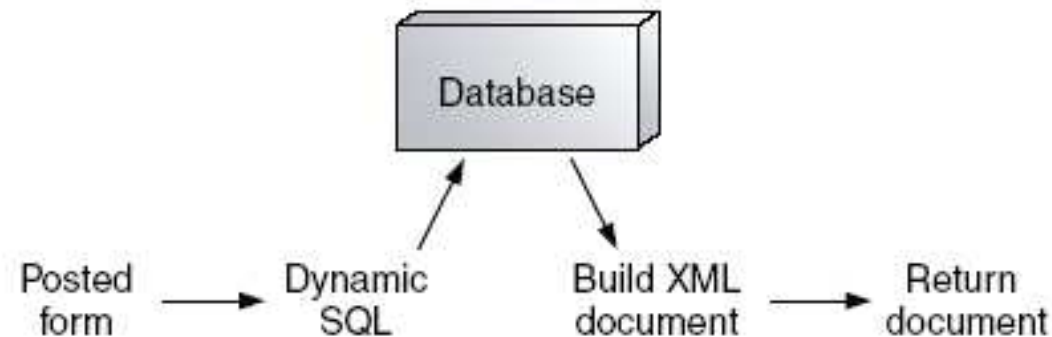
RIA: En la arquitectura cliente-servidor de una aplicación AJAX, la interacción dinámica con el servidor permite construir la presentación de la aplicación en respuesta a la navegación o entradas del usuario



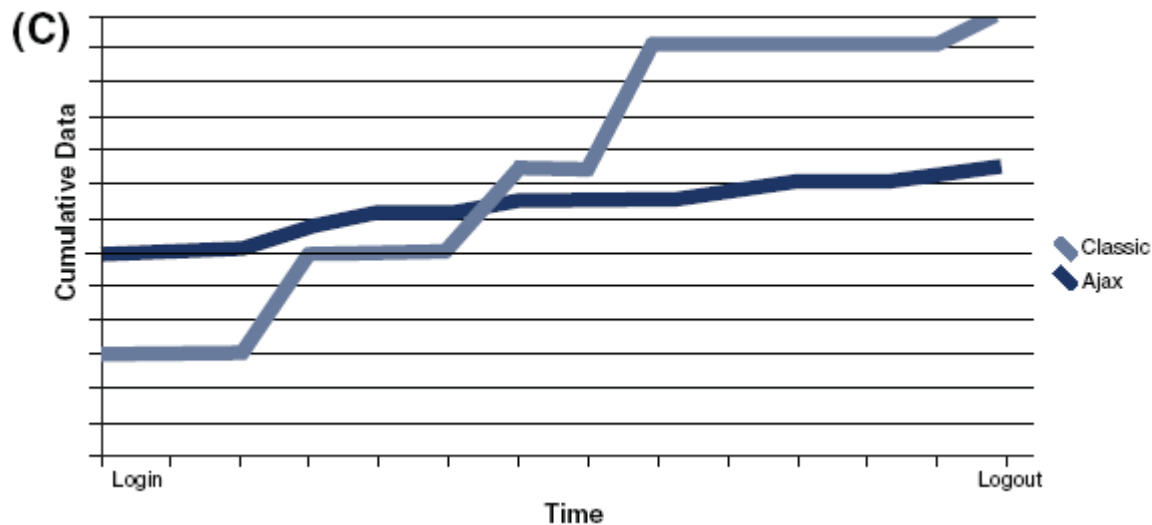
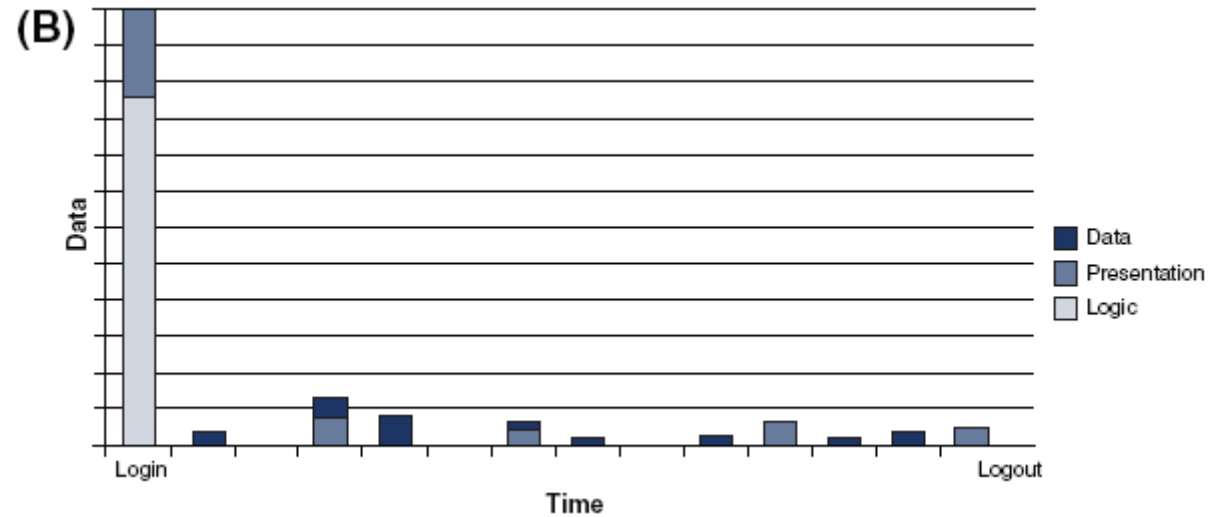
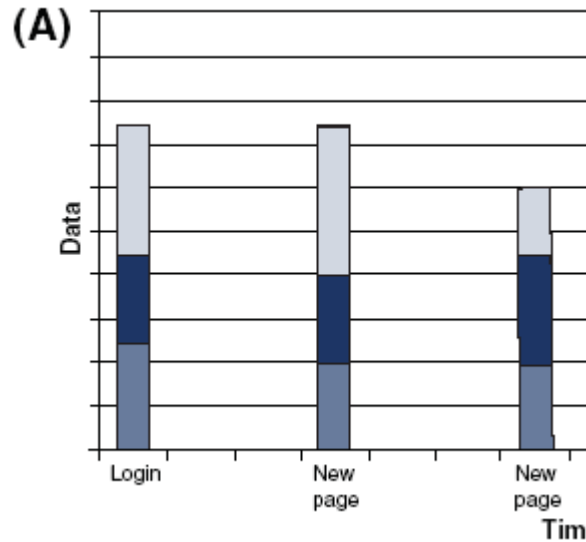
Consideraciones (3)



Datos estructurados: El acceso a los datos mediante SQL dinámicos permite construir documentos XML que mejoran su representación y manipulación por parte de la aplicación en el lado cliente



Consideraciones (4): Desempeño



(A) Tradicional
(B) AJAX
(C) Comparativo

Funcionamiento de componentes en AJAX

JavaScript define las reglas de negocio y el flujo del programa.

COM y CSS permiten a la aplicación reorganizar la apariencia en respuesta a los datos obtenidos en “background” desde el servidor por medio del objeto XMLHttpRequest

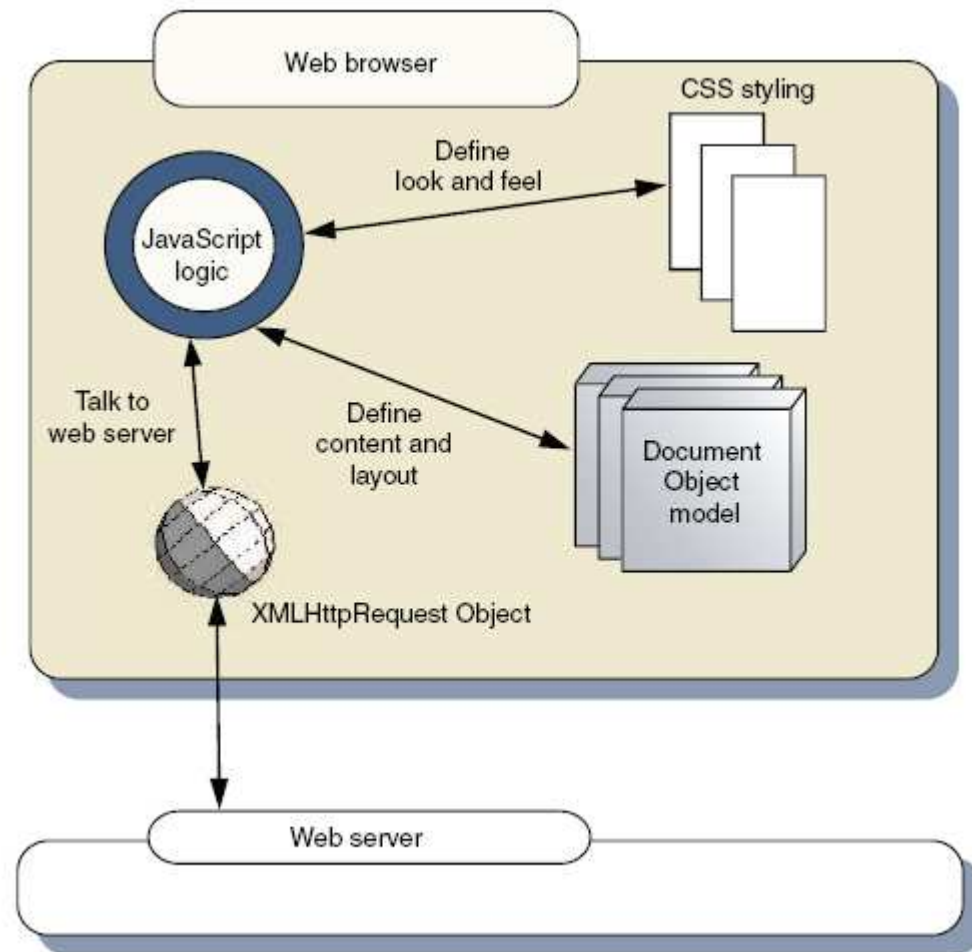
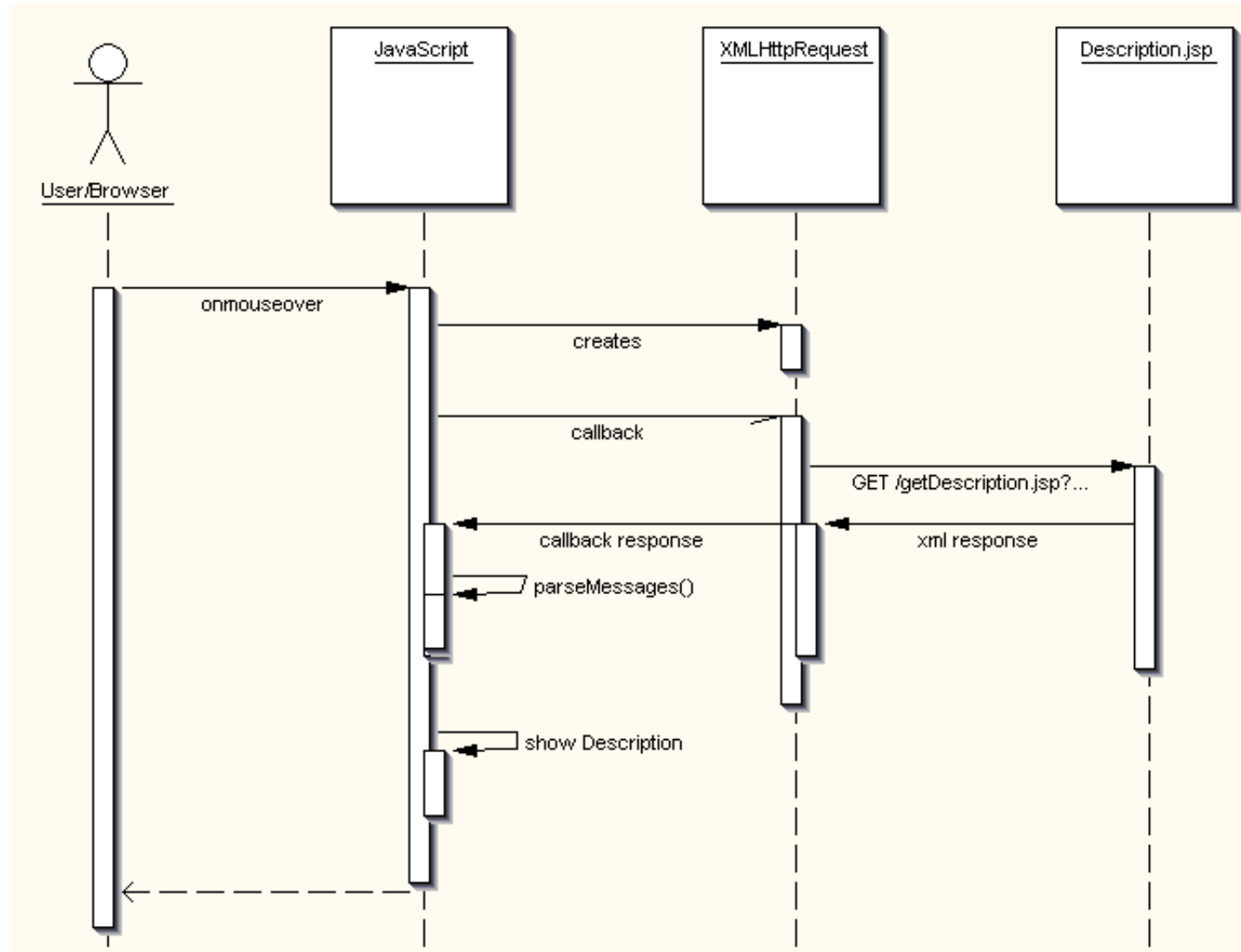


Diagrama de Secuencia de una invocación AJAX



El modelo DOM



Mantiene la estructura jerárquica de los datos contenidos en el browser, los cuales pueden ser modificados desde un lenguaje de *scripting*, como JavaScript

The screenshot shows a browser window with the address bar displaying 'file:///mnt/removable/writing/tech/ria/drafts/ch2/listing2.2a.html'. The developer tools are open, showing the DOM tree on the left and the Computed Style on the right. The DOM tree is expanded to show a `SPAN` element with the text 'hello' selected. The Computed Style panel shows various properties and their values, such as `background-color: transparent` and `color: rgb(255, 0, 0)`. The browser window at the bottom displays 'Hello World!' in red text on a green background.

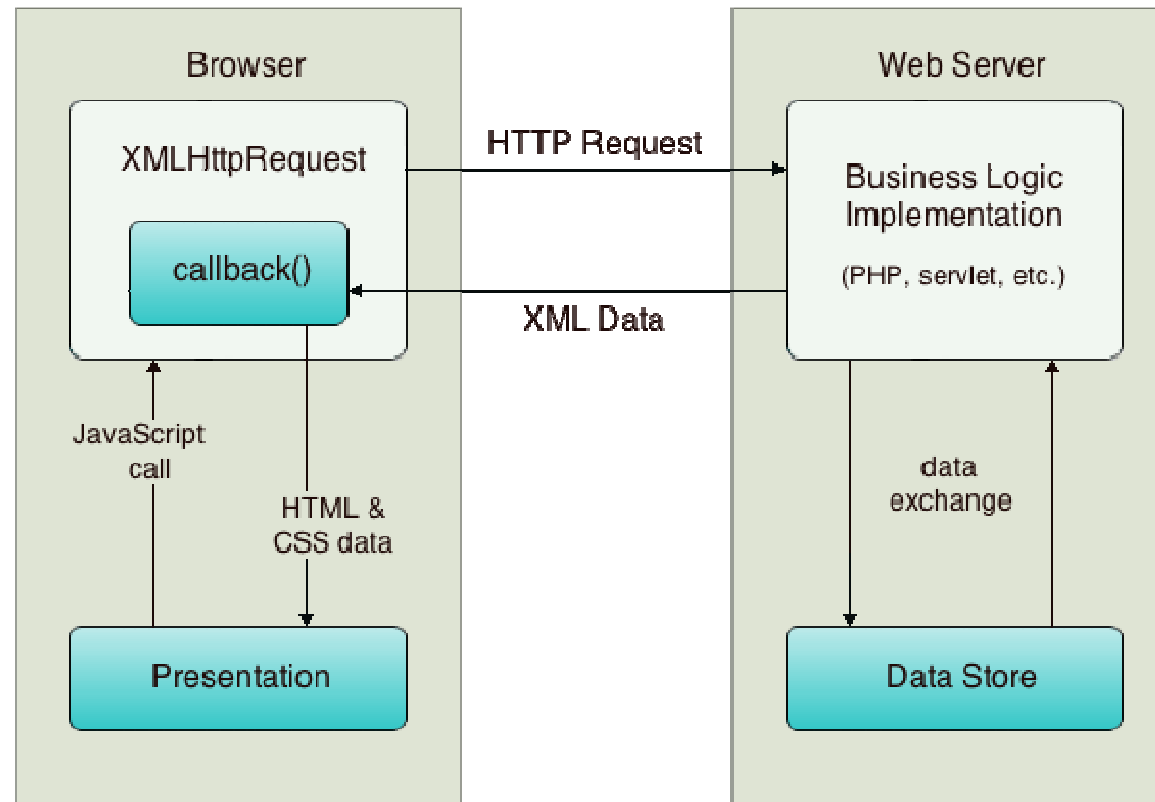
nodeName	id	class	Property	Value
#document			background-attachment	scroll
HTML			background-color	transparent
HEAD			background-image	none
#text			background-repeat	repeat
STYLE			border-bottom-color	rgb(255, 0, 0)
#text			border-bottom-style	none
#text			border-bottom-width	0px
BODY			border-collapse	separate
#text			border-left-color	rgb(255, 0, 0)
P			border-left-style	none
#text			border-left-width	0px
SPAN		hello	border-right-color	rgb(255, 0, 0)
#text			border-right-style	none
#text			border-right-width	0px
SPAN		world	border-top-color	rgb(255, 0, 0)
#text			border-top-style	none
#text			border-top-width	0px
#text			bottom	auto
#text			caption-side	top
#text			clear	none
#text			clip	auto
#text			color	rgb(255, 0, 0)

Manejo del asincronismo



- Se usa el objeto XMLHttpRequest para enviar la petición al servidor sin esperar respuesta inmediata del mismo
 - En algunos casos se usa XMLHttpRequest o iframes
- Se puede monitorear en que va la petición simultáneamente mientras se interactúa con el usuario en otros asuntos de la lógica de la aplicación del lado cliente

Arquitectura de una aplicación AJAX



Enviar petición al servidor



Solo se requiere pasarle el URL que generará los datos de vuelta al browser.

```
function sendRequest(url, params, HttpMethod) {
    if (!HttpMethod) {
        HttpMethod = "POST";
    }
    var req = getXMLHttpRequest();
    if (req) {
        req.open(HttpMethod, url, true);
        req.setRequestHeader
            ("Content-Type",
             "application/x-www-form-urlencoded");
        req.send(params);
    }
}
```


Monitoreo de la petición



Se implementa definiendo un punto de reentrada en el código para recoger los resultados devueltos por el servidor.

```
var READY_STATE_UNINITIALIZED=0;
var READY_STATE_LOADING=1;
var READY_STATE_LOADED=2;
var READY_STATE_INTERACTIVE=3;
var READY_STATE_COMPLETE=4;
var req;
function sendRequest(url, params, HttpMethod) {
    if (!HttpMethod){
        HttpMethod="GET";
    }
    req=getXMLHttpRequest();
    if (req){
        req.onreadystatechange=onReadyStateChange;
        req.open(HttpMethod,url,true);
        req.setRequestHeader
            ("Content-Type", "application/x-www-form-urlencoded");
        req.send(params);
    }
}
```

Monitoreo de la petición (2)



El punto de reentrada normalmente es una función de retorno (*callback function*) que será invocada cuando los resultado estén disponibles en el futuro.

```
function onReadyStateChange() {
    var ready=req.readyState;
    var data=null;
    if (ready==READY_STATE_COMPLETE) {
        data=req.responseText;
    }else{
        data="loading...["+ready+"]";
    }
    //... do something with the data...
}
```

Ejemplo Completo



```
<html>
<head>
<script type='text/javascript'>
var req=null;
var console=null;
var READY_STATE_UNINITIALIZED=0;
var READY_STATE_LOADING=1;
var READY_STATE_LOADED=2;

var READY_STATE_INTERACTIVE=3;
var READY_STATE_COMPLETE=4;
function sendRequest(url,params,HttpMethod){
  if (!HttpMethod){
    HttpMethod="GET";
  }
  req=initXMLHttpRequest();
  if (req){
    req.onreadystatechange=onReadyState;
    req.open(HttpMethod,url,true);
    req.setRequestHeader
      ("Content-Type", "application/x-www-form-urlencoded");
    req.send(params);
  }
}
```

Ejemplo Completo (2)



```
function initXMLHttpRequest(){
    var xRequest=null;
    if (window.XMLHttpRequest){
        xRequest=new XMLHttpRequest();
    } else if (window.ActiveXObject){
        xRequest=new ActiveXObject
            ("Microsoft.XMLHTTP");
    }
    return xRequest;
}
function onReadyState(){ ← Define callback handler
    var ready=req.readyState;
    var data=null;
    if (ready==READY_STATE_COMPLETE){ ← Check readyState
        data=req.responseText; ← Read response data
    }else{
        data="loading...["+ready+"]";
    }
    toConsole(data);
}
```

Initialize
request
object

Ejemplo Completo (3)



```
function toConsole(data){
  if (console!=null){
    var newline=document.createElement("div");
    console.appendChild(newline);
    var txt=document.createTextNode(data);
    newline.appendChild(txt);
  }
}
window.onload=function(){
  console=document.getElementById('console');
  sendRequest("data.txt");
}
</script>
</head>
<body>
<div id='console'></div>
```

```
</body>
</html>
```

SALIDA →

```
loading...[1]
loading...[1]
loading...[3]
Here is some text from the server!
```

Generación de datos (XML) para el cliente desde una base de datos



```
<?php
header("Content-type: application/xml");    ← Tell client we are returning XML
echo "<?xml version=\"1.0\" encoding=\"UTF-8\" ?>\n";
$db=mysql_connect("my_db_server", "mysql_user");
mysql_select_db("mydb", $db);
$sql="SELECT id,title,description,price,colors,sizes"
    ."FROM garments WHERE category=\"{$cat}\"";
$result=mysql_query($sql, $db);
echo "<garments>\n";
while ($myrow = mysql_fetch_row($result)) { ← Iterate through resultset
    printf("<garment id=\"%s\" title=\"%s\">\n"
        ."<description>%s</description>\n<price>%s</price>\n",
        $myrow["id"],
        $myrow["title"],
        $myrow["description"],
        $myrow["price"]);
    if (!is_null($myrow["colors"])){
        echo "<colors>{$myrow['colors']}</colors>\n";
    }
    if (!is_null($myrow["sizes"])){
        echo "<sizes>{$myrow['sizes']}</sizes>\n";
    }
    echo "</garment>\n";
}
echo "</garments>\n";
?>
```

Fetch the
results from
the database

Ejemplo de la salida XML



```
<garments>
  <garment id="SCK001" title="Golfers' Socks">
    <description>Garish diamond patterned socks. Real wool.
      Real itchy.</description>
    <price>$5.99</price>
    <colors>heather combo,hawaiian medley,wild turkey</colors>
  </garment>
  <garment id="HAT056" title="Deerstalker Cap">
    <description>Complete with big flappy bits.
      As worn by the great detective Sherlock Holmes.
      Pipe is model's own.</description>
    <price>$79.99</price>
    <sizes>S, M, L, XL, egghead</sizes>
  </garment>
</garments>
```

Algunos Frameworks AJAX



- [Dojo toolkit](#), Toolkit Modular JavaScript.
- [ExtJS](#), una librería que expande Prototype, JQuery y YUI.
- [jQuery](#), provee un framework Ajax y muchas otras utilidades.
- [Mootools](#), un framework compacto y modular mejor conocido por sus transiciones y efectos.
- [Prototype](#), provee framework Ajax y muchas otras utilidades.
- [qooxdoo](#), framework de aplicaciones Ajax. Es multiproposito e incluye un toolkit GUI.
- [Script.aculo.us](#), es utilizado con Prototype principalmente para animaciones y desarrollo de interfaces.

Alternativas para AJAX



- **Soluciones basadas en Macromedia Flash**
 - Macromedia's Flash es un sistema para desplegar películas interactivas mientras son descargadas desde el servidor
 - Se programa en Action-Script, un lenguaje muy cercano a JavaScript.
 - Provee soporte para formatos de entradas y controles.
 - Se usa en juegos hasta aplicaciones de negocios.
 - Aunque requiere un plug-in en el browser, tiene amplio soporte en muchos de ellos y en diferentes plataformas de sistemas operativos (Windows, Mac OS X, Linux,)
- **Java Web Start y tecnologías relacionadas**
 - Es una especificación para empaquetar aplicaciones java en un servidor web de forma que un proceso de escritorio las pueda encontrar, descargar y ejecutar
 - Viene con las JVM más recientes y la instalación la habilita automáticamente en los navegadores (Internet Explorer, Mozilla, etc.)
 - La aplicación se actualiza automáticamente si hay una nueva versión disponible
 - Se puede ejecutar la aplicación de forma desconectada lo cual reduce la carga sobre la red
 - Las aplicaciones son escritas casi siempre usando el Swing widget toolkit.

Créditos



- Dave Crane, Eric Pascarello, Darren James. Ajax in Action.
- A Relevance/Codecite Presentation