

TALLER DE PROGRAMACIÓN WEB

Sesión 08 - A

Fundamentos de AJAX

Computación e Informática
2017 - II

Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPRENDEDORES



LOGRO DE LA SESIÓN

En esta sesión aprenderás
qué es AJAX y como
funciona.



Computación e Informática
2017 - II

Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



DEFINICIÓN

- AJAX, acrónimo de **Asynchronous JavaScript And XML** (JavaScript y XML asíncronos, donde XML es un acrónimo de eXtensible Markup Language).
- Es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano.
- De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



DEFINICIÓN

- AJAX es la combinación de tres tecnologías ya existentes:
 - XHTML y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
 - Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones de ECMAScript como JavaScript, para mostrar e interactuar dinámicamente con la información presentada.
 - El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor web.
 - XML es el formato usado comúnmente para la transferencia devuelta por el servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Encuétranos en:

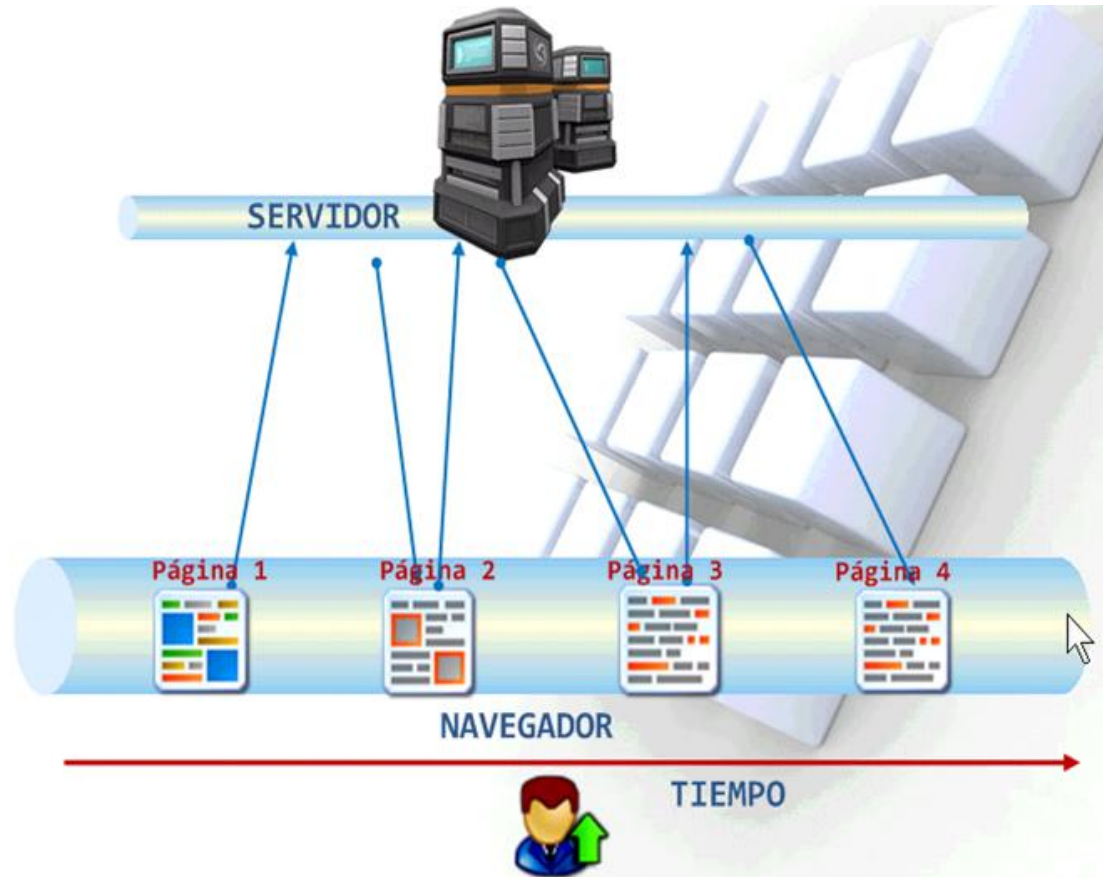


Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



CONVERSACIÓN TRADICIONAL CLIENTE-SERVIDOR



Encuétranos en:

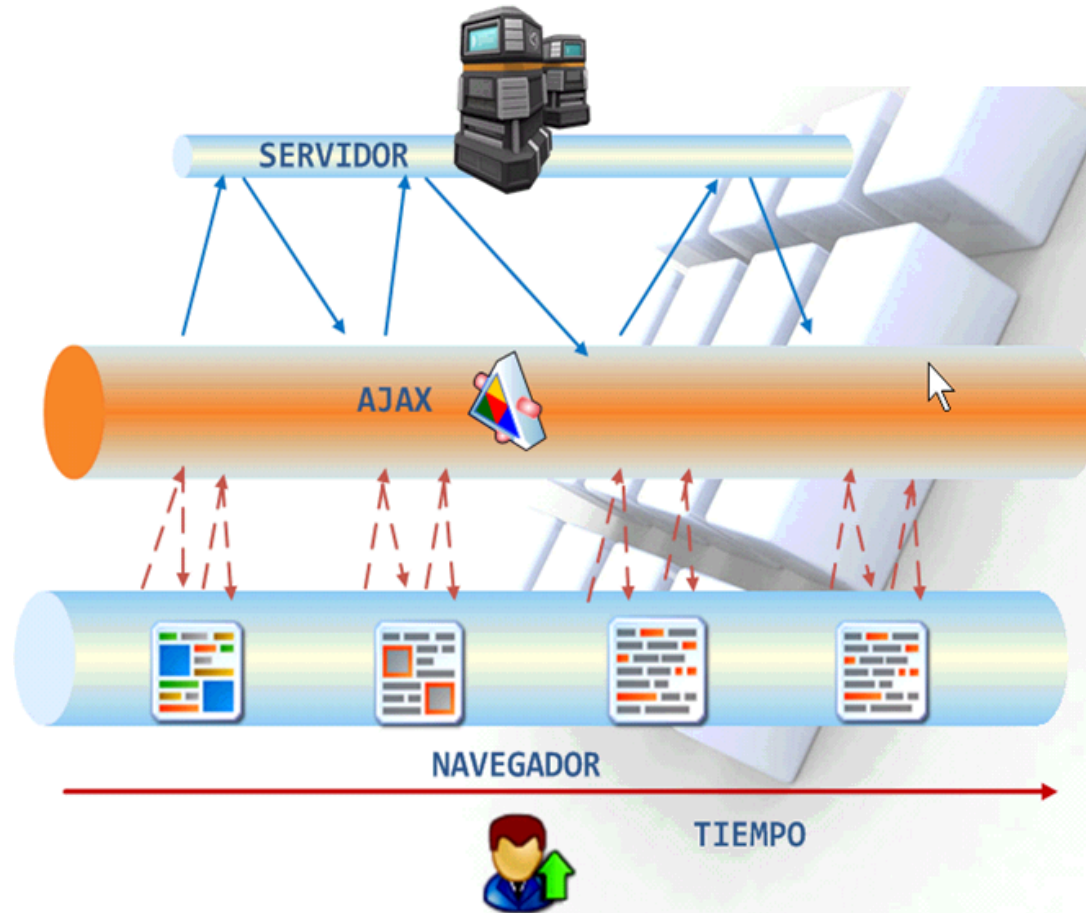


Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



INTERACCIÓN AJAX CLIENTE-SERVIDOR



Encuétranos en:

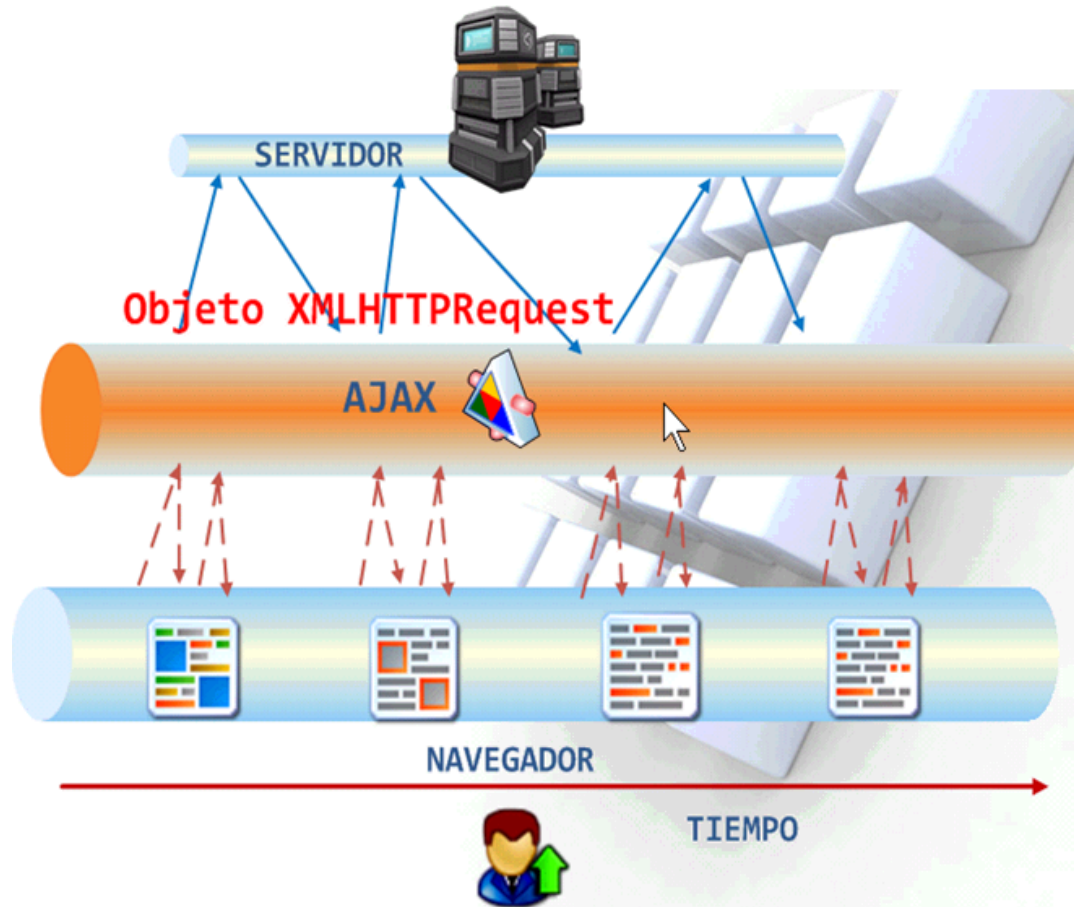


Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



INTERACCIÓN AJAX CLIENTE-SERVIDOR



Encuétranos en:

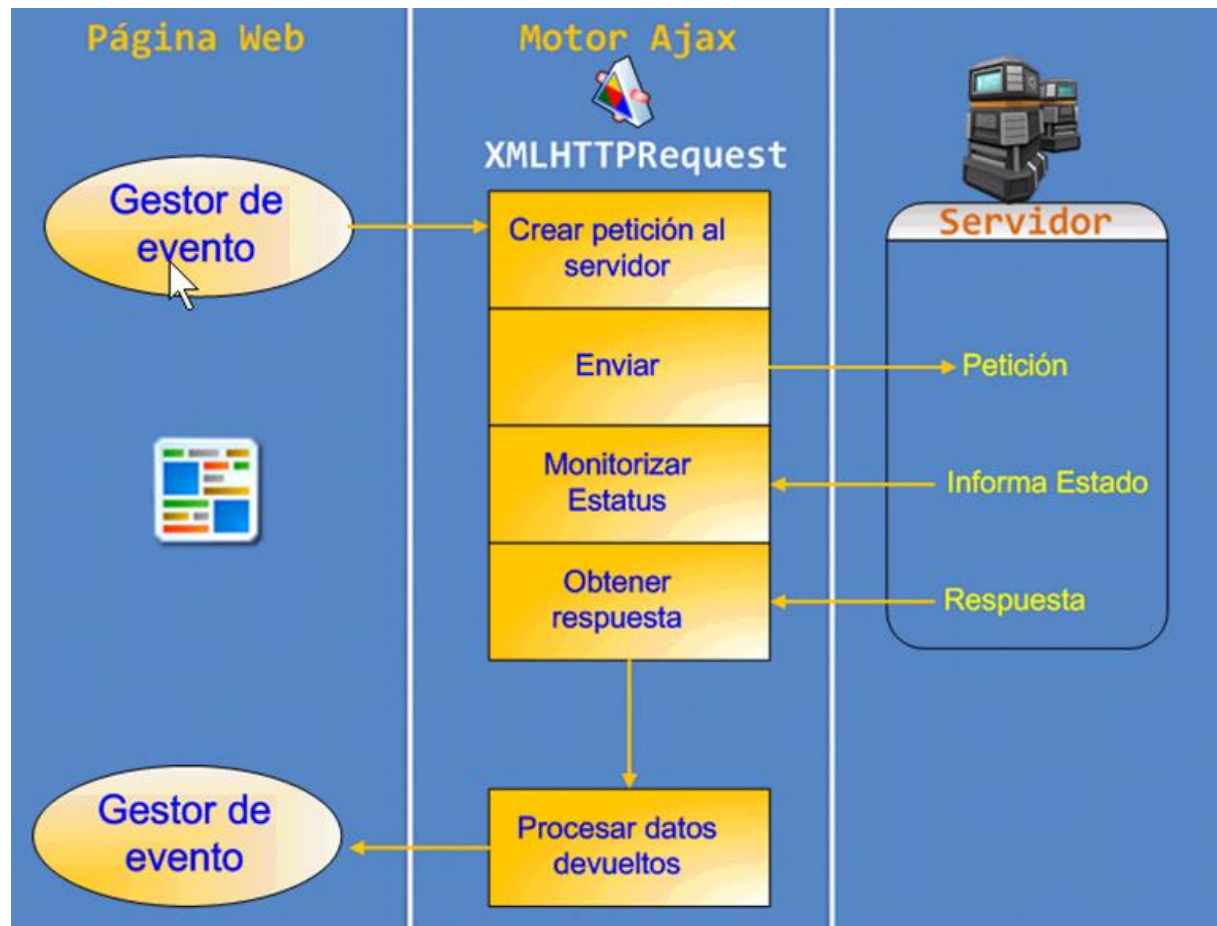


Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



INTERACCIÓN AJAX CLIENTE-SERVIDOR



Encuétranos en:

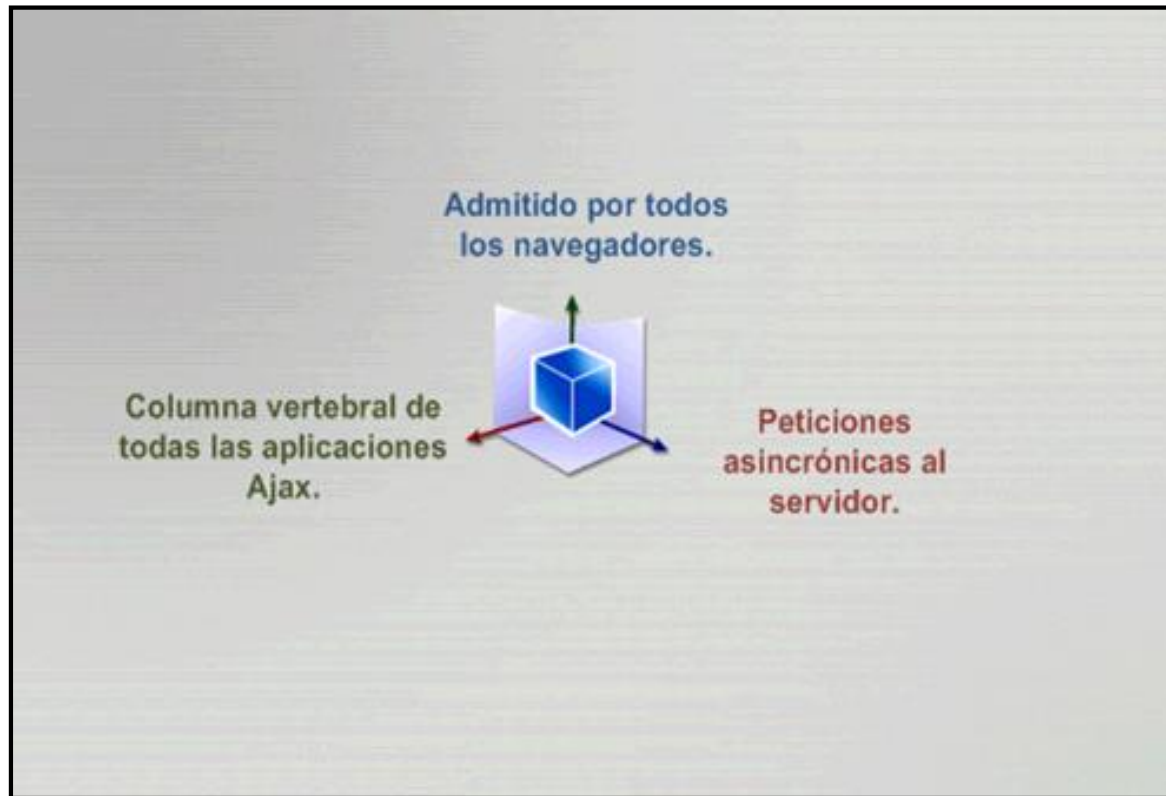


Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



EL OBJETO XMLHTTPREQUEST



Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



EL OBJETO XMLHTTPREQUEST



Encuétranos en:



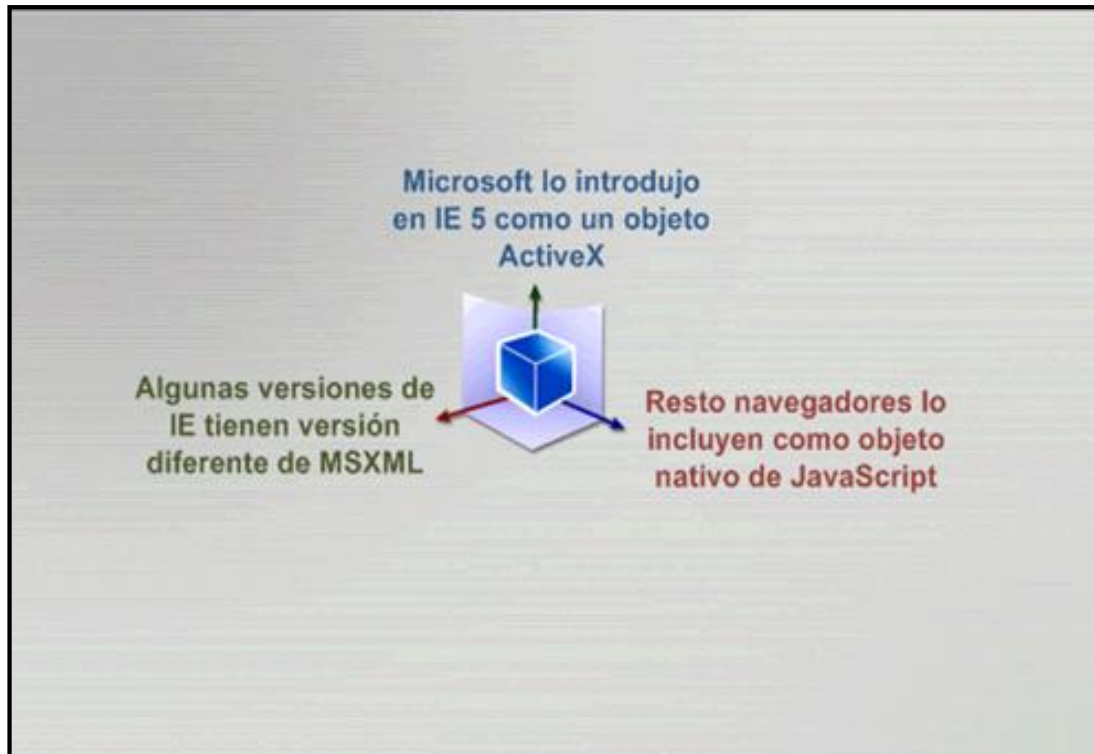
Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



EL OBJETO XMLHTTPREQUEST

Diferentes reglas para diferentes navegadores.



Encuétranos en:



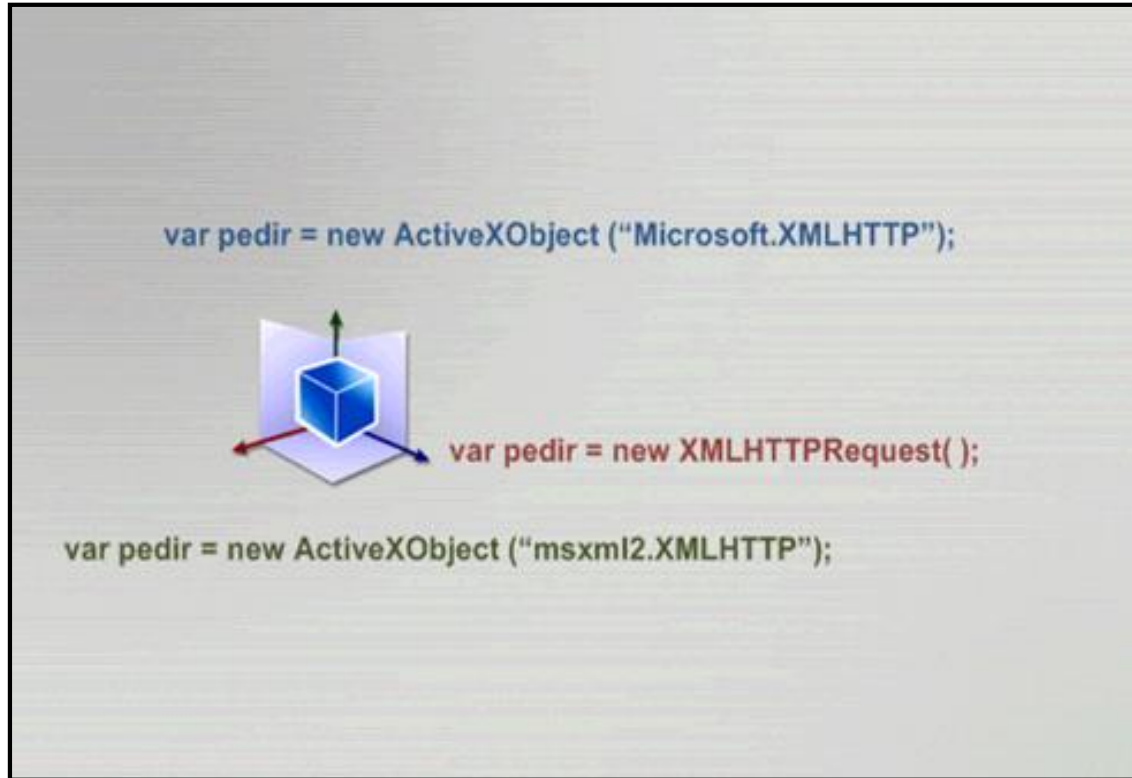
Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPRENDEDORES



EL OBJETO XMLHTTPREQUEST

Diferentes reglas para diferentes navegadores.



Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPRENDEDORES



EL OBJETO XMLHTTPREQUEST

Diferentes reglas para diferentes navegadores.

```
function getXMLHttpRequest() {  
    var req;  
    try {  
        req = new XMLHttpRequest();  
    } catch(err1) {  
        try {  
            req = new ActiveXObject("Msxml2.XMLHTTP");  
        } catch (err2) {  
            try {  
                req = new ActiveXObject("Microsoft.XMLHTTP");  
            } catch (err3) {  
                req = false;  
            }  
        }  
    }  
    return req;  
}
```

Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPRENDEDORES



EL OBJETO XMLHTTPREQUEST

Propiedades del objeto XMLHttpRequest

Propiedad	Descripción
onreadystatechange	Determina el gestor de evento que será llamado cuando la propiedad readyState del objeto cambie.
readyState	Número entero que indica el estado de la petición: 0: No iniciada 1: Cargando 2: Cargado 3: Interactivo 4: Completado
responseText	Datos devueltos por el servidor como una cadena de texto.
responseXML	Datos devueltos por el servidor como un objeto XMLHttpRequestDocument que se puede recorrer usando las funciones de JavaScript DOM.
status	Código de estado HTTP devuelto por el servidor.
statusText	Mensaje de texto enviado por el servidor junto al código (status), para el caso de código 200 contendrá "OK".

Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



EL OBJETO XMLHTTPREQUEST

Métodos del objeto XMLHttpRequest

Propiedad	Descripción
abort()	Detiene la petición actual.
getAllResponseHeaders()	Devuelve todas las cabeceras como una cadena.
getResponseHeader(etiqueta)	Devuelve el valor de la etiqueta en las cabeceras de la respuesta.
open(método,URL,asíncrona)	Especifica un método HTTP (GET o POST), la URL objetivo, y si la petición debe ser manejada asincrónicamente (true o false).
send(contenido)	Envía la petición, opcionalmente con datos POST.
setRequestHeader(etiqueta,valor)	Establece el valor de una etiqueta de las cabeceras de petición.

Encuétranos en:



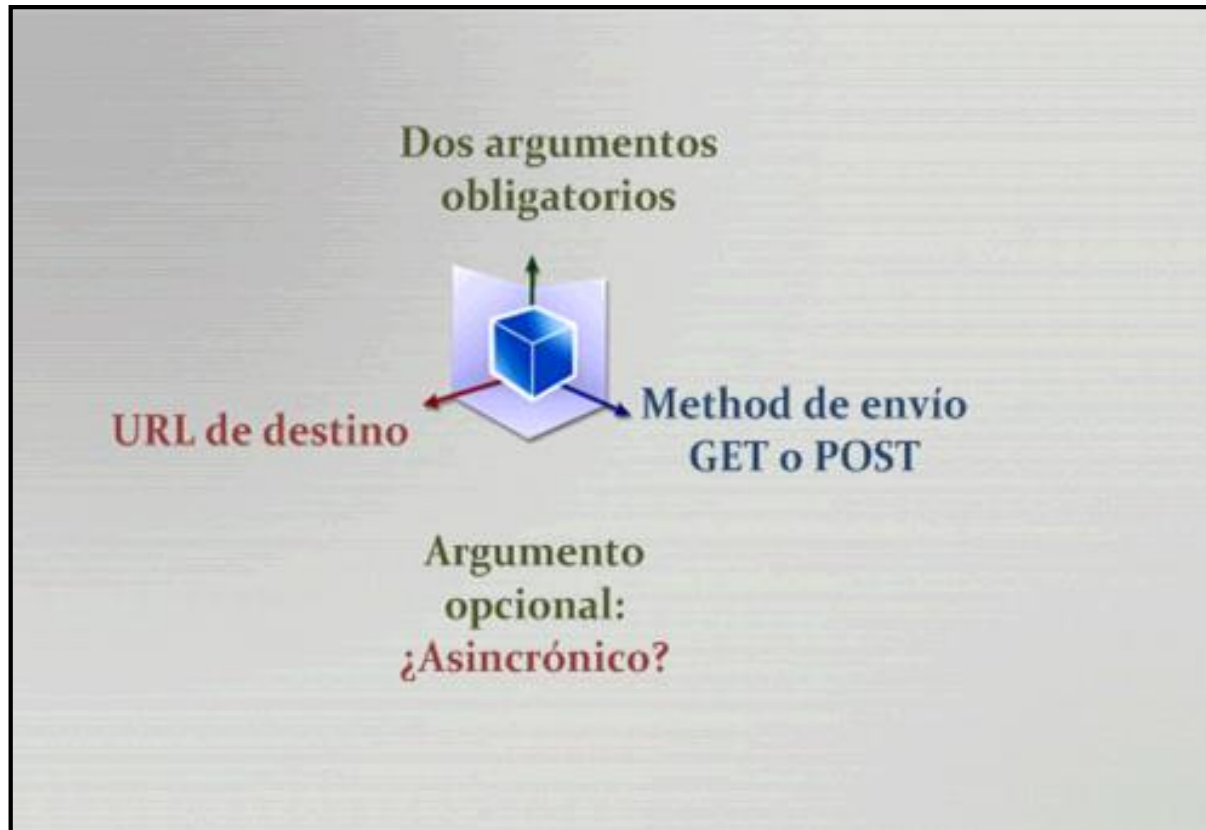
Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



EL OBJETO XMLHTTPREQUEST

Método: open(método, URL, asíncrona)



Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



EL OBJETO XMLHTTPREQUEST

Método: send(contenido)



Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



IMPLEMENTACIÓN

Ejemplo 01

En este ejemplo se está enviando los datos utilizando el método GET y se esta recibiendo un texto.

procesar.jsp

```
<%  
// Datos  
int n1 = Integer.parseInt( request.getParameter("n1") );  
int n2 = Integer.parseInt( request.getParameter("n2") );  
// Proceso  
int suma = n1 + n2;  
// Reporte  
out.print(suma);  
%>
```

formulario.html

Suma de Dos Números

Número 1	<input type="text"/>
Número 2	<input type="text"/>
Suma	<input type="text"/>
<input type="button" value="Calcular"/>	<input type="button" value="Limpiar"/>

www.desarrollasoftware.com

Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENDEDORES



IMPLEMENTACIÓN

Ejemplo 02

En este ejemplo se está enviando los datos utilizando el método POST, se esta recibiendo un documento XML y la función de procesamiento se esta construyendo en línea.

procesar.jsp

```
<%@page contentType="text/xml" %>
<%
// Datos
int n1 = Integer.parseInt( request.getParameter("n1") );
int n2 = Integer.parseInt( request.getParameter("n2") );
// Proceso
int suma = n1 + n2;
// Reporte
out.print( "<rpta><suma>" );
out.print( suma );
out.print( "</suma></rpta>" );
%>
```

formulario.html

Suma de Dos Números

Número 1

Número 2

Suma

Calcular

Limpiar

www.desarrollasoftware.com

Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPRENEDORES



IMPLEMENTACIÓN

Ejemplo 03

En este ejemplo se está enviando los datos utilizando el método POST, luego se retorna el resultado en formato **JSON** (Arreglo) y la función de procesamiento está utilizando la función **eval()** para convertir el resultado en un arreglo JavaScript.

procesar.jsp

```
<%  
// Datos  
int n1 = Integer.parseInt( request.getParameter("n1") );  
int n2 = Integer.parseInt( request.getParameter("n2") );  
// Proceso  
int suma = n1 + n2;  
// Reporte  
out.print( "[" + suma + "]" );  
%>
```

Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

formulario.html

Suma de Dos Números

Número 1

Número 2

Suma

Calcular

Limpiar

www.desarrollasoftware.com



INSTITUTO DE
EMPREENDEDORES



IMPLEMENTACIÓN

Ejemplo 04

En este ejemplo se está enviando los datos utilizando el método POST, luego se retorna el resultado en formato **JSON** (Object) y la función de procesamiento está utilizando la función **eval()** para convertir el resultado en un objeto JavaScript.

procesar.jsp

```
<%  
// Datos  
int n1 = Integer.parseInt( request.getParameter("n1") );  
int n2 = Integer.parseInt( request.getParameter("n2") );  
// Proceso  
int suma = n1 + n2;  
// Reporte  
out.print( "{ valor:" + suma + "}" );  
%>
```

formulario.html

Suma de Dos Números

Número 1

Número 2

Suma

Calcular

Limpiar

www.desarrollasoftware.com

Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPRENDEDORES



IMPLEMENTACIÓN

Ejemplo 05

En este ejemplo se está enviando los datos utilizando el método POST y retorna una porción de código HTML que es visualizado en un **div** de nombre **divResultado**.

formulario.html

Suma de Dos Números

Número 1

Número 2

Resultado

Número 1	17
Número 2	23
Suma	40

www.desarrollasoftware.com

procesar.jsp

```
<%  
// Datos  
int n1 = Integer.parseInt(request.getParameter("n1"));  
int n2 = Integer.parseInt(request.getParameter("n2"));  
// Proceso  
int suma = n1 + n2;  
%>  
<h2>Resultado</h2>  
<table width="190">  
  <tr>  
    <td width="82">Número 1</td>  
    <td width="96"><%=n1%></td>  
  </tr>  
  <tr>  
    <td>Número 2</td>  
    <td><%=n2%></td>  
  </tr>  
  <tr>  
    <td>Suma</td>  
    <td><%=suma%></td>  
  </tr>  
</table>
```

Encuétranos en:



Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

INSTITUTO DE
EMPREENEDORES



INSTITUTO DE
EMPREENDEDORES

