



---

## AJAX: Análisis Tecnológico

---

Facultad de Ciencias de la Computación

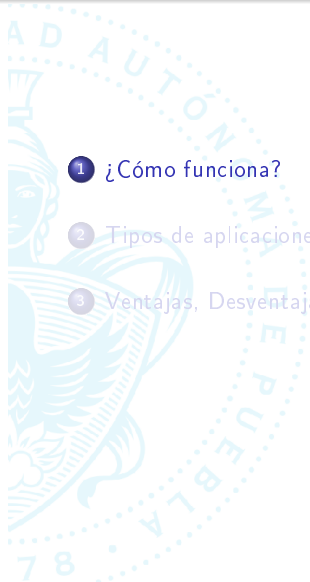
Juan Carlos Conde R.

*Web Technologies*

# Contenido

- 1 ¿Cómo funciona?
- 2 Tipos de aplicaciones Web
- 3 Ventajas, Desventajas y Alternativas

# Contenido

- 
- 1 ¿Cómo funciona?
  - 2 Tipos de aplicaciones Web
  - 3 Ventajas, Desventajas y Alternativas

# Una “amalgama” de Tecnologías, I

Contrariamente a lo que se pudiera creer, **AJAX NO es una tecnología específica e innovadora, sino más bien una conjunción de varias tecnologías viejas.**

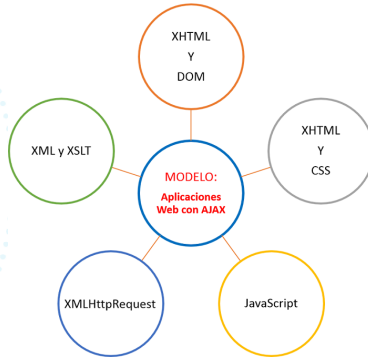
Así, las aplicaciones AJAX usan todo o partes de las siguientes tecnologías:

# Una “amalgama” de Tecnologías, II

- ▶ **CSS** (*Cascading Style Sheets*) que permiten aplicar un formato en el contenido de una página XHTML.
- ▶ **DOM** (*Document Object Model*) que representa la jerarquía de los elementos de una página XHTML.
- ▶ **XMLHttpRequest** como objeto que asegura las transferencias asíncronas (o síncronas) entre cliente y servidor.
- ▶ **XML** y **JSON** para las transferencias de datos entre el servidor y el cliente.
- ▶ **JavaScript** que para el scripting e interacción entre estas tecnologías.

## Una “amalgama” de Tecnologías, III

El interés de AJAX es utilizar estas diferentes tecnologías es que ya están integradas en los navegadores más recientes.



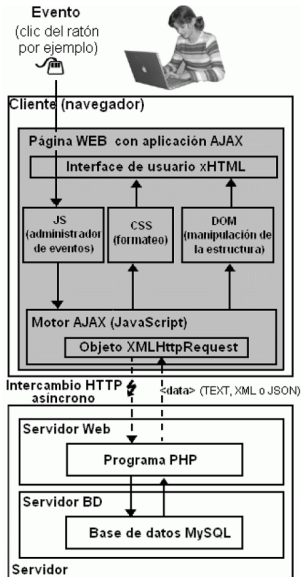
Por lo tanto son inmediatamente explotables, pero con algunas restricciones de implementación que subsisten de un navegador a otro.

# Los cimientos de AJAX, I

JavaScript es la tecnología ideal, del lado del cliente, para cumplir con la misión de interactuar con todas estas tecnologías.

- ▶ Permite crear el “motor” del sistema de comunicación donde la transferencia de datos se realiza con la ayuda del objeto XMLHttpRequest.
- ▶ En lo que se refiere al intercambio de datos, puede utilizar diferentes formatos (xml, json, text) según la complejidad de los flujos de información.
- ▶ La explotación de las respuestas del servidor actúan sobre:
  1. CSS para modificar el formato de la página XHTML.
  2. DOM para modificar el contenido o la estructura de la página XHTML.

# Los cimientos de AJAX, II



- Evento generado por el usuario (clic en el ratón por ejemplo)
- Solicitud HTTP enviada al servidor (generada por el navegador)
- <data> Datos devueltos en respuesta por el servidor (TEXT, XML o JSON)
- Intercambio HTTP Sincrónico
- Intercambio HTTP Asincrónico



# Contenido

- 1 ¿Cómo funciona?
- 2 **Tipos de aplicaciones Web**
- 3 Ventajas, Desventajas y Alternativas

# Funcionamiento de una aplicación Web estática, I

En esta parte del documento, vamos a comparar el funcionamiento de una aplicación AJAX con un sitio Web estático y el de un sitio Web dinámico.

Funcionamiento de una aplicación Web estática:

- ▶ Con un sitio Web estático, la única interactividad de la que dispone el usuario es pasar de una página HTML a la otra por un simple clic en los enlaces de hipertexto presentes en una página.
- ▶ Cuando el usuario hace clic en un vínculo, se envía una solicitud HTTP, estableciendo al mismo tiempo una comunicación con el servidor.

## Funcionamiento de una aplicación Web estática, II

- ▶ Esta comunicación es de tipo síncrono, es decir que desde la emisión de la solicitud, la comunicación permanece en su lugar hasta la recepción de la respuesta del servidor.
- ▶ Durante el tiempo de tratamiento de la petición, el navegador se queda “congelado”, bloqueando así cualquier acción posible del usuario.
- ▶ Para cada solicitud, el servidor devolverá una respuesta en forma de una página HTML completa.

## Funcionamiento de una aplicación Web estática, III

- ▶ Si se trata de una simple petición, tras la escritura por parte del usuario de la URL específica de una página en la barra de direcciones del navegador o, más comúnmente, cuando el usuario da clic en un vínculo hipertexto, el servidor sólo devolverá la página HTML solicitada, lo que cerrará el tratamiento del lado del servidor y desbloqueará así al navegador.



# Funcionamiento de una aplicacion Web dinámica, I

Anteriormente vimos el tratamiento de una simple petición al servidor pero se pueden existir otros casos, por ejemplo: durante el envío de un formulario.

Para el caso de una aplicación Web dinámica, lo que permitirá al servidor procesar correctamente la información es una consulta que consta de:

1. una línea de solicitud (especificando el método utilizado y el protocolo HTTP),
2. un cuerpo (que contiene los datos enviados al servidor en el caso de una solicitud emitida con el método POST) y
3. una serie de encabezados que definen los detalles de la consulta (naturaleza del navegador utilizado, codificación...).

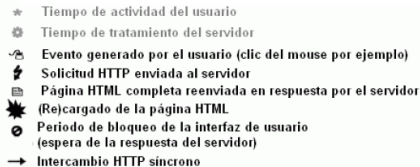
## Funcionamiento de una aplicacion Web dinámica, II

Por lo general, al enviar un formulario, el tratamiento del lado del servidor se realiza por una página que contiene un programa (en PHP por ejemplo).

Los datos recibidos pueden manejarse directamente por el programa o hacer un intercambio con un servidor de base de datos con el fin de memorizarlas o emitir una consulta SQL.

Al final de este tratamiento, se construirá una nueva página HTML en la ejecución y reenviada al explorador, que concluirá el proceso, liberando al navegador de la misma manera que con un sitio estático.

## Diagrama de tiempo de una aplicación Web dinámica tradicional



# Funcionamiento de una aplicacion Web con AJAX, I

Una vez cargado el motor de AJAX en el navegador esperará el evento para el cual ha sido programado.

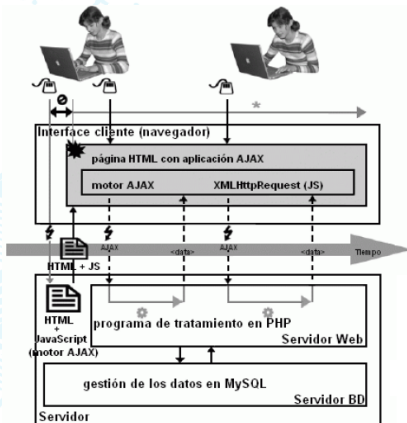
- ▶ Para esto, un controlador de eventos JavaScript se configura para llamar el motor desde la aparición misma del evento considerado.
- ▶ Durante la llamada del motor, un objeto XMLHttpRequest se instancia y luego se configura, enseguida se envía una solicitud asíncrona al servidor.



## Funcionamiento de una aplicacion Web con AJAX, II

- ▶ Tras la recepción del mismo, el servidor comenzará su tratamiento y devolverá la respuesta HTTP correspondiente.
- ▶ Esta última será considerada por la función de llamado del motor AJAX que explotará los datos para mostrarlos en un lugar específico de la pantalla.
- ▶ Una de las principales diferencias entre una aplicación Web tradicional y una aplicación AJAX es el intercambio asíncrono de datos entre el navegador y el servidor.

## Diagrama de una aplicación Web con AJAX en modo *asíncrono*



- \* Tiempo de actividad del usuario
- ⚙️ Tiempo de tratamiento del servidor
- 🖱️ Evento generado por el usuario (clic en el ratón por ejemplo)
- ⚡ Solicitud HTTP enviada al servidor (generada por el navegador)
- 🔗 Solicitud HTTP enviada al servidor (generada por el motor AJAX)
- <data> Datos reenviados en respuesta por el servidor (TEXT, XML o JSON)
- ☀️ Cargado de la página HTML
- 🕒 Período de bloqueo de la interfaz de usuario
- Intercambio HTTP Sincrónico
- > Intercambio HTTP Asíncrono

# Contenido

- 
- 1 ¿Cómo funciona?
  - 2 Tipos de aplicaciones Web
  - 3 Ventajas, Desventajas y Alternativas**

# Ventajas y Desventajas de AJAX, I

## Ventajas:

- ▶ Evita la recarga de la página complete y economiza el ancho de banda.
- ▶ Evita el bloqueo de la aplicación durante el tratamiento de la solicitud.
- ▶ Aumenta la capacidad de respuesta de la aplicación.
- ▶ Mejora de la ergonomía de la interfaz.

## Ventajas y Desventajas de AJAX, II

### Desventajas:

- ▶ No hay memorización de las acciones en el historial del navegador.
- ▶ Falta de indexación de contenidos dinámicos en motores de búsqueda y barra de Favoritos.
- ▶ Dependencia de la activación de JavaScript en el navegador.

## Alternativas a AJAX

Existe una técnica llamada “marcos ocultos” (`iframe`) usada mucho antes del objeto XMLHttpRequest; permite establecer las comunicaciones en *background* con el servidor y que, como AJAX, no requieren agregar un *plug-in*.

`iframe` (*inline frame* o marco incorporado)

Es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal. Fue introducido en Internet Explorer en el '97 y durante mucho tiempo sólo fue soportado en este navegador. Actualmente la etiqueta `iframe` es aceptada por la W3C como un elemento estándar y es ampliamente soportada por una gran variedad de navegadores.

## Los marcos ocultos, I

Esta técnica aprovecha la estructura de marcos HTML, ya que uno de ellos es invisible y sirve como un puente para comunicarse con el servidor.

El marco oculto se hace invisible al establecer su anchura y altura en cero píxeles.

Con esta técnica, es posible enviar solicitudes al servidor sin interrumpir la pantalla del usuario.

## Los marcos ocultos, II

Para ilustrar el funcionamiento de esta técnica, detallaremos el ciclo de una comunicación completa entre el navegador y el servidor.

1. Para empezar, el usuario activa una función JavaScript de la estructura visible.
2. Esta función llamara a un *script* del servidor cuyo retorno se asignará al marco oculto.



## Los marcos ocultos, III

3. El *script* del servidor analiza entonces los parámetros proporcionados y procesa la solicitud.
4. Luego reenvía en respuesta al marco oculto una página HTML completa que contiene el resultado en una etiqueta `<div>`.
5. En esta misma página HTML se encuentra una función JavaScript que se invocará cuando la página se carga completamente dentro del marco oculto (con el controlador de evento `window.onload` por ejemplo).

## Los marcos ocultos, IV

6. Finalmente, cuando la función JavaScript se ejecuta en el marco oculto, esta recupera el resultado insertado previamente en la etiqueta `<div>` de la misma página y lo asigna a un área definida del marco visible.
7. El usuario puede entonces ver la respuesta que aparece en la página visible del navegador y esto sin dejar de utilizar la interfaz durante el tratamiento por parte del servidor evitando que la página se vuelva a cargar.



The diagram consists of a large green square containing a smaller, slightly offset green square. Inside the smaller square, the text is as follows:

```
<iframe>  
object  
embed
```

# Ventajas y Desventajas de los marcos ocultos

## Ventajas:

- ▶ Funciona en los navegadores antiguos.
- ▶ Mantiene el historial del navegador. Esto permite usar botones *Siguiente* y *Anterior*.

## Desventajas:

- ▶ Falta de información sobre el estado de la solicitud HTTP realizada en *background*.

# Consideraciones del uso de marcos ocultos

Nótese que algunas aplicaciones acoplan a AJAX la técnica de los marcos ocultos.

Es posible programar un retraso temporal para interrumpir el tratamiento e informar al usuario sobre el estado de la solicitud HTTP.

Sin embargo, es preferible utilizar XMLHttpRequest para mantener el control de todas las etapas del tratamiento de la solicitud.

# Comentarios...

Juan Carlos Conde R.  
[carlos.conder@correo.buap.mx](mailto:carlos.conder@correo.buap.mx)