

AJAX Mobile – Interfaces de usuario para Mobile Web 2.0

Pedro Ballesteros Herranz y José Luís Núñez Díaz
Telefónica I+D
C/ Emilio Vargas, 6. 28043 Madrid
Teléfono: 913374000, Fax: 913374004
Correo Electrónico: pmbh@tid.es y jln@tid.es

Resumen

Es un hecho que la Web 2.0 está significando para Internet casi un renacimiento. Es el motor que da nuevo impulso a Internet, proporcionando una nueva generación de servicios en los que la comunidad de usuarios vuelve a depositar su confianza. La Web 2.0 está cambiando aspectos que afectan a la percepción de la red a nivel social y tecnológico, así como a sus modelos de negocio.

En el plano tecnológico se están sustituyendo las clásicas páginas web estáticas por verdaderas aplicaciones de usuario final ejecutadas directamente en el navegador, gracias sin duda alguna a la tecnología AJAX. La adopción de AJAX se presenta como un buen candidato para facilitar una “movilización” de la Web 2.0.

1. Introducción

La industria de las aplicaciones de datos en el mundo móvil es un terreno del que se puede decir que aun está en fase de crecimiento. No ha sido capaz de penetrar de forma amplia entre los usuarios finales.

La operadoras reportan que aun existiendo millones de unidades habilitadas para operar con estas tecnologías, los ingresos anuales relacionados con estos servicios representan, en comparación, un bajo porcentaje. Los servicios de este tipo más ampliamente utilizados acaban concentrándose en servicios de distribución de contenidos (descarga de melodías, imágenes, vídeos, etc.).

Lo que plantea este artículo es cómo conseguir una mayor penetración de Internet en el mundo móvil capitalizando los factores que explican el éxito de las aplicaciones etiquetadas como Web 2.0 a la hora de “reenganchar” a los usuarios a Internet en el mundo fijo. Utilizando los mismos conceptos que protagonizan la discusión sobre Web 2.0, no pretendemos ni incluir las redes móviles en el mapa de la Web 2.0, ni crear uno nuevo para Mobile Web 2.0, sino **aprender del modelo aplicado por las killer applications de la Web 2.0** para vitalizar la Internet móvil como ellas hicieron en la Internet fija.

Por otro lado hay que destacar que esta Mobile Web 2.0 no se reduciría tan solo a la replicación de la Web 2.0 fija. Aplicada a un entorno movilizado podría tener incluso mayor potencial. Se podría enriquecer con servicios que solo están disponibles en un entorno móvil, como ubicuidad, localización y posicionamiento, seguridad, etc. (Ej. *Mashup*¹ de *Google Maps* combinado con servicios de localización).

Por tanto, cuando hablamos de Mobile Web 2.0, nos referimos a esa experiencia de aplicar los principios de la Web 2.0 para vitalizar la Internet Móvil.

De entre todos los principios que rigen la Web 2.0, vamos a analizar en detalle aquel que considera la **web como plataforma**, prestando especial atención a AJAX como la tecnología que consigue su resolución de una manera más exitosa.

1.1. Acerca de Web 2.0

En la Internet fija, después de superado el estallido de la burbuja tecnológica del año 2001, se está consiguiendo un nuevo impulso en el que están jugando un papel muy importante las aplicaciones encuadradas dentro de la Web 2.0, que animan tanto a usuarios como a proveedores a depositar de nuevo su confianza en la web.

Las principales compañías de Internet están adaptando sus portales y sitios a los conceptos de Web 2.0. Servicios como el *weblog* y la sindicación por RSS son ya un requisito que cualquier sitio Web debe incluir. De hecho, opiniones expresadas y publicadas en la *blogosfera* trascienden en ocasiones a medios de comunicación ajenos a Internet como la televisión o la prensa escrita, e incluso llegan a afectar al entorno político, lo que demuestra la importancia creciente que tienen estos movimientos en la sociedad actual. Tecnologías como AJAX se están introduciendo en todos los portales, ofreciendo aplicaciones listas para ejecutarse desde el navegador, interactivas y sin barreras de portabilidad y distribución, contribuyendo a una mejora en la experiencia de los usuarios y acercándonos más a la idea de “un solo cliente universal para todas las aplicaciones”.

El concepto de Web 2.0 se define mediante el desarrollo de los siguientes siete principios:

1. La web como plataforma.
2. Aprovechamiento de la inteligencia colectiva.
3. Los datos son el impulso conductor.

¹ *Mashup*: Uno de los muchos elementos que componen la Web 2.0. Sitio web o aplicación web que usa contenido de más de una fuente para crear un nuevo servicio completo.

4. Fin del ciclo de versionado del software a través de la beta perpetua.
5. Modelos de programación ligeros.
6. Software no limitado a un sólo dispositivo.
7. Experiencia del usuario enriquecida.

Se trata de identificar cómo se están aplicando estos principios en las aplicaciones Web 2.0, para a su vez **identificar como afectarían al mundo móvil**, es decir, sobre aplicaciones que tienen que ejecutarse en un entorno diferenciado y con sus propias particularidades como es el de un dispositivo móvil.

1.2. Acerca de AJAX

Aunque la Web 2.0 se describe generalmente mediante el desarrollo de sus siete principios, de una forma detallada se puede observar que en realidad está compuesta por una nube de elementos tanto tecnológicos como conceptuales (Fig. 1-1).

Uno de los elementos tecnológicos que aparece en esta nube aportando un valor de gran peso es AJAX (*Asynchronous JavaScript And XML*). Aunque AJAX se presenta como una tecnología “opcional” para la Web 2.0, lo cierto es que precisamente esta tecnología es la que está jugando un papel fundamental en el impulso de la Web 2.0.

La mayor parte de las aplicaciones que componen la Web 2.0 se apoyan en esta tecnología. AJAX es lo que permite a los navegadores el acceso directo a unas aplicaciones de usuario final con interfaces tan dinámicas y enriquecidas que poco tienen que envidiar a muchas aplicaciones nativas de escritorio.

AJAX potencia algunos de los principios que definen la Web 2.0: concibe **la web como plataforma, no se limita a un solo dispositivo**, y contribuye tanto a crear una **experiencia de usuario enriquecida** como al fin del versionado a través de **la beta perpetua**.

AJAX no es la única solución para conseguir este tipo de comportamiento, pero lo cierto es que actualmente se presenta como la solución más realista en cuanto a las posibilidades de los navegadores y sus problemas de compatibilidad. Su uso está cada vez más extendido y por tanto su validez está probada. Es la única tecnología que no acaba desembocando en el artificio de la instalación de plug-ins que amplían las capacidades y tecnologías soportadas por los navegadores, lo que termina por convertirse en contribuciones adicionales a la ya existente fragmentación del mercado y la proliferación de nuevas barreras para la distribución.

1.3. AJAX y Mobile Web 2.0

La apariencia poco atractiva de las aplicaciones de la Internet móvil, así como los lenguajes de etiquetado utilizados para la navegación, se han considerado tradicionalmente como importantes barreras que frenan una amplia aceptación por parte de los usuarios finales.

La pregunta es, “¿puede AJAX jugar el mismo papel para enriquecer en el móvil las interfaces de las aplicaciones y de esta forma hacer la Internet móvil más atractiva para los usuarios?”.

Una de las ventajas de apoyarse en AJAX para la introducción de Web 2.0 en el entorno móvil, es que las tecnologías a introducir no son invenciones nuevas, ni tecnologías de última generación. AJAX utiliza tecnologías existentes desde hace años de forma nativa en los navegadores, como DHTML, CSS, DOM, JavaScript, etc. Aportando la ventaja adicional de que se dispone desde el principio de una amplia comunidad de desarrolladores.

Esta optimización en el uso de antiguas tecnologías, se puede considerar como la solución que se ha visto forzada a adoptar la comunidad de desarrolladores

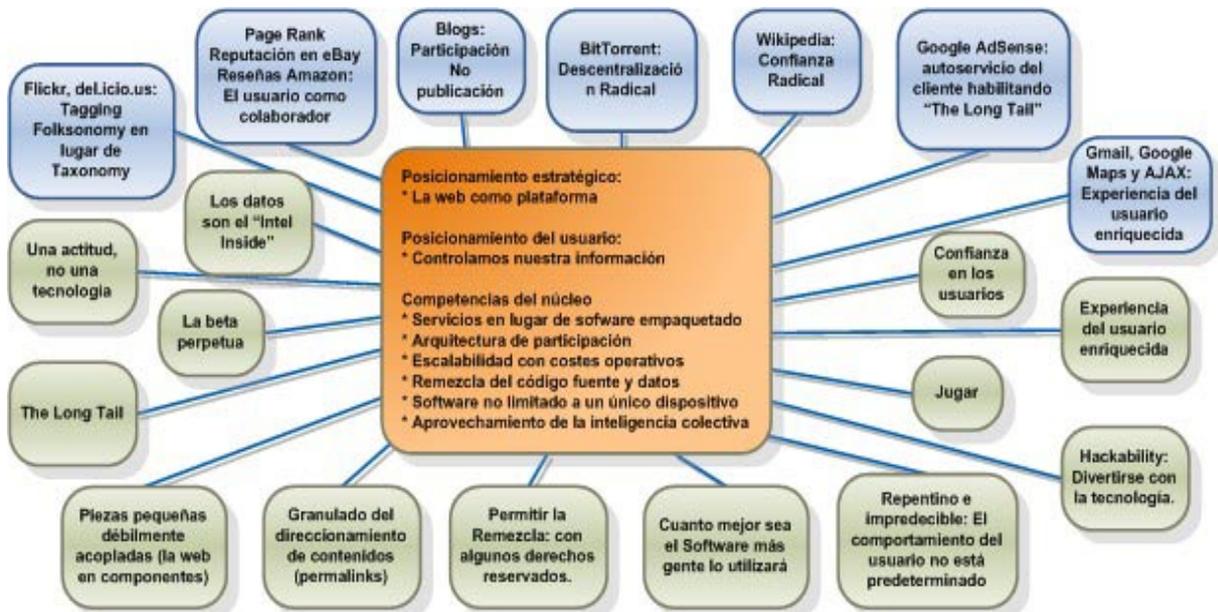


Figura 1-1: Mapa Mental de la Web 2.0 propuesto por Tim O'Reilly para la descripción de la Web 2.0.

para la creación de aplicaciones interactivas en la web. Ha sido la respuesta a una guerra de tecnologías entre navegadores que impedía la adopción de una solución común.

En la Internet móvil se está produciendo esta misma falta de acuerdo entre los fabricantes de navegadores. Por tanto, la adopción de AJAX en el móvil, que realmente sería la adopción de tecnologías propias de Internet, puede convertirse en la alternativa común que permita la creación de una web móvil interactiva, igual que lo ha sido para la Internet fija.

AJAX se presenta como un buen candidato para replicar en el entorno de la Internet Móvil la misma revolución que se está produciendo en el entorno de la Internet Fija. AJAX es la tecnología que puede acelerar la “movilización” de la Web 2.0.

2. Web 2.0

La Web 2.0 no es ni una tecnología ni la nueva versión de algo tan etéreo como Internet. Aludiendo al sistema de numeración utilizado para designar las actualizaciones del software, el término Web 2.0 hace referencia a un grupo de conceptos, tendencias o actitudes que caracterizan una nueva generación de servicios y aplicaciones existentes en Internet, y que representan una clara evolución desde la web estática tradicional. Manteniendo la misma analogía, se suele utilizar el término Web 1.0 para designar todo lo que se queda fuera de Web 2.0, pero no por ello se convierte en obsoleto o deja de estar vigente en Internet.

No existe una definición simple y concreta para Web 2.0, ya que ni siquiera existe un acuerdo totalmente aceptado sobre su significado. Describir de forma completa la Web 2.0 acaba siempre por convertirse en un extenso análisis de los servicios y aplicaciones que la originaron, así como su impacto a escala social, económica y tecnológica.

Esta falta de concreción se entiende al observar que no ha sido la definición o especificación de una Web 2.0 lo que ha posibilitado una nueva generación de servicios y aplicaciones, ha sido precisamente a la inversa. El concepto de Web 2.0 ha nacido y se ha moldeado desde servicios y aplicaciones que ya formaban parte de Internet y tenían algunos rasgos distintivos respecto de aplicaciones previas.

El concepto de Web 2.0 se originó en un *brainstroming* realizado entre O'Reilly y MediaLive International, analizando como había sido la evolución de ciertos servicios significativos de Internet (Fig. 2-1). Observando los principios que gobernaban esta evolución se dio lugar al *Mapa Mental de la Web 2.0* (Fig. 1-1) que engloba como una nube todos los conceptos que describen o forman parte de la Web 2.0 y que muestran que no existe una clara frontera entre lo que es y lo que no es Web 2.0. En realidad la Web 2.0 toma forma

mediante la aplicación de un conjunto de principios y prácticas que giran en torno a unos conceptos centrales.

Los resultados finales de estas primeras reuniones se presentaron en la primera “*Web 2.0 Conference*” de Octubre del 2004, donde se describió la Web 2.0 en base a los siete principios clave que caracterizan sus aplicaciones, de los cuales consideramos los más significativos desde el plano conceptual y desde el plano tecnológico los siguientes:

- *La inteligencia colectiva*: La web se convierte en una red social que se nutre de contenidos mediante una cultura de participación, y potencia modelos de negocio como “*The Long Tail*”.
- *La web como plataforma*: Las aplicaciones web tradicionales se convierten en aplicaciones RIA (*Rich Internet Applications*) ejecutadas directamente en el navegador.

2.1. La inteligencia colectiva

El principio del aprovechamiento de la inteligencia colectiva es quizás el que mejor resume las claves de Web 2.0, y el que mejor dicta el camino a seguir en la creación de servicios y aplicaciones en Internet móvil.

Este principio se basa en una consolidación del concepto de las *Redes Sociales*, con la que se produce una transformación en la manera de crear, distribuir y consumir los contenidos de la red.

En la web tradicional el usuario actúa como un mero espectador. Los creadores de los portales deciden cual es el contenido más atractivo, y a los usuarios sólo se les queda la capacidad para consumirlos, sin ningún control sobre éstos.

El éxito de servicios como el *weblog* o las *wikis* demuestran que el usuario demanda un mayor nivel de participación. El usuario comparte información y



Figura 2-1: Ejemplo de evolución de Web 1.0 hacia una Web 2.0 propuesta por Tim O'Reilly como primera interpretación de lo que significa la Web 2.0.

colabora en la creación de contenidos. Se crean comunidades virtuales con las que se genera una inteligencia colectiva que dirige de forma espontánea la creación de contenidos. En esta cultura de la participación son los propios usuarios de la red los que se dedican a generar contenidos que pueden ser valiosos para otros usuarios.

Por ejemplo, un *blog* introduce algunos sencillos elementos que potencian el modelo de la página personal a través del principio de participación. La organización cronológica de un *blog*, lejos de ser una diferencia trivial, conduce a una cadena de reparto, de publicidad y de valor completamente distinta. La posibilidad de que los lectores puedan publicar comentarios sobre cada artículo introduce otro elemento adicional de participación. La sindicación por RSS (*Really Simple Syndication*) eleva a un nuevo nivel el *bookmark* o marcador de los navegadores. No sólo permite enlazarse con una página concreta, también suscribirse a la misma con notificaciones cada vez que la página cambia. La creación de enlaces permanentes (“*permalinks*”) para cada entrada individual del *blog* hace relativamente fácil discutir directamente acerca de una entrada concreta del sitio.

Actualmente se estima que el número de *blogs* en todo el mundo supera los 30 millones y que se viene multiplicando por dos cada seis meses.

La *Wikipedia* es un ejemplo de radical confianza basada en la inverosímil idea de que cualquier usuario puede añadir una nueva entrada, y esta puede ser modificada por cualquier otro. Actualmente la *Wikipedia* contiene más de 1,4 millones de artículos en inglés, unos 160.000 en español, y otros tantos en otros idiomas.

Los servicios de compartición de recursos como *You.Tube*, *Flickr* o *Del.icio.us*, muestran ejemplos en los que se potencia el concepto de “*folksonomy*” en contraste con “*taxonomy*”. La semántica de la web también se crea a través de un ejercicio de colaboración y confianza. La participación en la clasificación de los contenidos mediante el uso de etiquetas libremente elegidas por los usuarios.

Otras redes sociales como *MySpace*, *Second life* o *Tagworld* permiten la creación de verdaderas comunidades virtuales. *MySpace* por ejemplo facilita la creación de una red de amigos, la publicación y compartición de contenidos, la carga de ficheros e incluso la creación de *blogs*.

El éxito que proporciona esta cultura de la participación marca una pauta a seguir. Cualquier servicio debería proporcionar al usuario, en mayor o menor medida, un canal que le permita participar de forma activa, en definitiva, **interactuar**. El usuario encuentra atractiva la posibilidad de **participar**, **colaborar** y **compartir**.

Así se encuentra que incluso negocios que poca relación pueden tener con esta inteligencia colectiva, ofrecen de alguna forma algún tipo de participación. *Amazon* facilita la publicación de opiniones sobre los productos comprados, lo que abre líneas activas de participación. *Ebay* también ofrece una forma de participación mediante su mecanismo de valoración de compradores y vendedores.

Un ejemplo de negocio con éxito que aprovecha esta cultura de la participación es el servicio *AdSense* de *Google*, con una publicidad basada más en la colaboración que en la espectacularidad de la misma, utilizando sencillos anuncios de texto y evitando los formatos de publicidad preferidos por los publicistas.

AdSense se basa en que el éxito de la publicidad en *Internet* está en la posibilidad de ofrecer anuncios de interés para cada usuario concreto, y en la capacidad para auto-distribuirse mediante la confianza en la cultura de la participación. Es decir, ahora deciden los propios consumidores, no los publicistas.

La publicidad de *Google* ofrece un mecanismo con el que cualquier sitio puede incluir sus anuncios a cambio de una participación en los beneficios. La selección de los anuncios mostrados se realiza de forma automática mediante el análisis semántico del contenido de las páginas en las que se incrustan. Los visitantes obtienen una publicidad acorde con sus intereses.

Abordando el paradigma de la participación dentro de la *Mobile Web 2.0* se plantea una pregunta, ¿aporta la movilidad algo más a este escenario de colaboración?, pues precisamente la ubicuidad del acceso a los servicios. El usuario tiene más flexibilidad para colaborar en esta inteligencia colectiva. Participación en cualquier momento y desde cualquier sitio, sin la necesidad de un punto fijo de acceso.

2.2. The Long Tail

El éxito de la publicidad de *Google* también fue fruto de la comprensión del modelo “*The Long Tail*”. El poder colectivo de los sitios web pequeños conforman la gran mayoría del contenido en la Web.

El concepto de “*Long Tail*” también marca una pauta a seguir en los modelos de negocios. El nombre tiene origen en la gráfica de la distribución

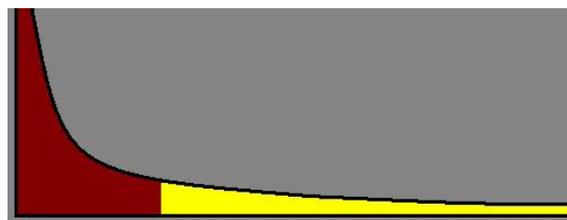


Figura 2-2: *The Long Tail*. El conjunto total de productos poco demandados (la cola) puede equiparar al total de los productos de más éxito.

de la población de consumidores (*Fig 2-2*). En esta se muestra que la mayor demanda se concentra en el 20% de los productos (los más populares), mientras que el 80% restante se reparte entre los productos menos demandados. El modelo tradicional, limitado por los costes de distribución y almacenaje, se basaba en la ley de Pareto o regla del 80/20 donde se recomendaba centrarse en el 20% de los productos y usuarios para obtener el 80% de los ingresos.

Los nuevos modelos de negocio proponen explotar este 80% de productos poco demandados, ya que si el canal de comunicación o distribución es suficientemente amplio, la demanda en conjunto de todos estos productos menos populares puede equiparar la demanda de los productos de éxito. La Web 2.0 fortalece este modelo de negocio, la cultura de la participación facilita la difusión de artículos poco conocidos a una amplia audiencia.

De esta forma *Amazon* estima que obtiene entre el 25% y 30% (el 57% según algunos estudios) de sus ingresos de la venta de los libros que no son las habituales superventas.

Otro ejemplo de éxito que incluye la estrategia *long tail*, atendiendo a la demanda de lo popular, lo menos popular y lo raro, es de nuevo la publicidad de *Google*, al descubrir que la *long tail* configura un mercado publicitario interesante. Existen unas pocas palabras muy buscadas, pero la mayoría de las búsquedas son muy diferentes, por tanto *Google* no dirigió su atención a sacar beneficio de las pocas búsquedas muy frecuentes (o relacionadas con sitios conocidos), como hace la publicidad tradicional, si no que se centró en la gran variedad de las búsquedas poco frecuentes, diseñando un modelo para los publicistas pequeños que se dirigen a mercados pequeños (la cola).

2.3. La web como plataforma.

La tendencia actual borra cada vez más la frontera entre el ordenador y la red. Siguiendo esta línea Macromedia introdujo en el año 2002 el concepto de **Rich Internet Application** (RIA), aplicaciones web que presentan la funcionalidad y las características de las tradicionales aplicaciones de escritorio.

Las aplicaciones RIA potencian la interfaz de usuario de las aplicaciones de Internet, mediante un **enriquecimiento y un aumento de su interactividad**, mostrando las siguientes características principales:

- Ejecución en un navegador sin necesidad de instalación de nuevo software.
- Ejecución local en un entorno seguro (*sandbox*).
- Comunicación asíncrona con el servidor para la transferencia de la información.

Las aplicaciones de Internet enriquecidas forman parte de los conceptos de la Web 2.0. La creación de

aplicaciones que puedan utilizarse desde la red, pero que siempre han sido consideradas aplicaciones de escritorio, tales como procesadores de texto, hojas de cálculo, etc. Pasar de un modelo de instalación donde la red no pasa de ser un mero canal de transmisión, a un modelo de ejecución, donde la red es precisamente la plataforma de ejecución.

La web tradicional ha estado siempre dominada por el modelo de interacción natural al protocolo HTTP y a la publicación HTML. Es un modelo síncrono y poco interactivo. Los navegadores son utilizados tan solo para la presentación más o menos estática de la información. El mayor problema de este modelo es que toda la interacción con la aplicación debe pasar por un modelo de petición-espera-respuesta. El usuario desencadena peticiones al servidor, mientras se procesan se debe esperar y finalmente se recibe una respuesta que sustituye a la página actual.

En el acceso a una aplicación RIA se realiza una primera carga en la que el navegador obtiene lo que se llama el "*client engine*". Es la capa que desde entonces actúa como intermediario entre la interfaz de usuario y el servidor. El usuario accede a una interfaz que replica en el navegador la funcionalidad habitual de las aplicaciones de escritorio, pasando de una navegación por hipervínculos a una verdadera interactividad. La interfaz de usuario no se vuelve a recargar. Cualquier acción del usuario que requiera comunicación con el servidor se resolverá mediante una petición asíncrona en la que se suele utilizar XML, el *client engine* analizará los datos recibidos y modificará lo mínimo indispensable de la interfaz. El usuario nunca provoca recargas de nuevo HTML, puede seguir utilizando la aplicación mientras las comunicaciones se resuelven en segundo plano.

Los beneficios proporcionados por las aplicaciones RIA son numerosos:

- Desde cualquier ubicación geográfica el usuario puede acceder tanto a sus datos como a las aplicaciones con las que se manipulan, independientemente del equipo utilizado.
- La actualización y distribución de aplicaciones es instantánea. La nueva funcionalidad se hace inmediatamente accesible (la beta perpetua).
- La demanda de recursos de computación entre clientes y servidores está mejor balanceada. Los clientes pueden asumir parte de la lógica que tradicionalmente se ejecuta en el servidor.
- La comunicación asíncrona con los servidores aumenta la sensación de interactividad, ofreciendo mejores tiempos de respuesta y mejorando la experiencia del usuario.
- Aumenta la eficiencia de la red al reducir el volumen de datos intercambiados durante el uso de la aplicación. Ya no se transfieren elementos de presentación en cada acción del usuario, tan solo se transmiten datos.

- Aumento de la seguridad del equipo local ya que el usuario no necesita instalar aplicaciones que aumenten el riesgo de virus.

Existen numerosas tecnologías que hacen realidad el concepto de aplicaciones enriquecidas (*Flash*, *ActiveX*, *applets Java*, etc.), algunas de las cuales existen desde los inicios de Internet. Sin embargo, AJAX, al utilizar tecnologías nativas del navegador no requiere la instalación de *plugins*, cuando el usuario es cada vez más reactivo a la instalación de componentes y más dispuesto a la ejecución directa de las aplicaciones. AJAX propone una nueva forma de utilizar esas tecnologías que posibilita la creación de aplicaciones y no sólo el maquetado de la apariencia de una información estática.

Pero el motivo del éxito de AJAX podría definirse también como el único motivo de su uso. Como se comenta más adelante, desde un plano puramente técnico AJAX es la tecnología que mayores dificultades plantea, y la menos preparada para desarrollar aplicaciones RIA, siendo superada en muchos aspectos por el resto de tecnologías.

Sin embargo, la Web 2.0 se está poblando de forma creciente con cada vez más aplicaciones basadas en AJAX. Encontrando los ejemplos más representativos en *Google*, que ha sido el mayor promotor de esta tecnología, con aplicaciones como *GMail*, *Google Maps*, *Google Calendar*, o el espectacular *Google Docs* (la reciente compra de *Google* del anterior *Writely*), verdadero procesador de textos en línea.

3. La tecnología AJAX

Actualmente AJAX es la tecnología que, de forma más pragmática, permite a los usuarios el uso de aplicaciones que emulan la misma interfaz y usabilidad que tienen las aplicaciones nativas, pero ejecutadas un navegador sin *plugins* adicionales, es decir, acceder a aplicaciones disponibles en la red sin pasos previos de descarga e instalación.

Las principales contribuciones de AJAX a la Web 2.0 pueden resumirse en:

1. El usuario utiliza aplicaciones en red que se ejecutan en el navegador, sin necesidad de descarga e instalación previa.
2. Las aplicaciones sólo interactúan con la red para intercambiar la información que están procesando, nunca su apariencia.
3. El usuario accede a la misma aplicación, desde cualquier equipo de la red, con cualquier plataforma nativa de ejecución.
4. El proveedor de la aplicación no distribuye nuevas versiones, sino que según va aumentando la funcionalidad de la aplicación en la red, estará inmediatamente disponible para todos los usuarios. Se elimina el ciclo del versionado del software (*beta perpetua*).

Asynchronous JavaScript And XML no es una única tecnología. Se trata en realidad de una técnica que combinando un conjunto de tecnologías, existentes de forma nativa en los navegadores, permite la **creación de aplicaciones interactivas** y directamente ejecutables en la web (aplicaciones RIA).

Existen muchas formas diferentes de utilizar las técnicas AJAX, así como diferentes posibilidades en cuanto a la selección de las tecnologías utilizadas. La mayor parte de las aplicaciones AJAX se apoyan en las siguientes: *JavaScript*, *HyperText Markup Language (HTML)*, *Cascading Style Sheets (CSS)*, *Document Object Model (DOM)*, *Extensible Markup Language (XML)* y el objeto *XMLHttpRequest*.

JavaScript es el lenguaje de programación que se utiliza para implementar una aplicación AJAX.

Es el encargado de mantener unidos el resto de elementos, definiendo el flujo de trabajo y la lógica de negocio de la aplicación en la parte cliente.

La apariencia de la interfaz de usuario se crea mediante la combinación de HTML y CSS. Pero a diferencia de una aplicación web tradicional, en AJAX la mayor parte de este HTML+CSS se genera en el navegador usando *JavaScript*, en lugar de obtenerlo desde el servidor.

El DOM del navegador exporta la página web como una estructura de objetos programables que pueden ser manipulados con *JavaScript*. La manipulación de este DOM permite a una aplicación AJAX modificar partes de la página directamente en el navegador.

La combinación *HTML+CSS+DOM* forma lo que se conoce como *Dynamic HTML (DHTML)*, que existe desde hace tiempo en todos los navegadores.

En las técnicas AJAX se hace uso muy intensivo de DHTML para la creación páginas web que simulan la interfaz de usuario de las aplicaciones nativas, tanto en su apariencia como en su funcionalidad, permitiendo incluso la implementación de acciones como la de arrastrar y soltar (*drag and drop*).

El uso de *JavaScript* para modificar y generar el HTML a través del DOM es una posibilidad que existe en los navegadores desde mucho antes de AJAX, pero nunca había sido explotada de una forma tan amplia.

Aunque esta combinación de tecnologías ya proporciona la capacidad para crear interfaces de usuarios enriquecidos en el navegador, aun falta por introducir un elemento de interacción con el servidor que le proporcione utilidad real. De nada serviría una interfaz basada en controles funcionales o *widgets* si cada vez que el usuario iniciara una acción se tuviera que volver a recargar toda la página web. Es necesaria una comunicación asíncrona que permita al *JavaScript* progresar los

elementos visuales de la interfaz en base a los datos intercambiados.

El objeto *XMLHttpRequest* accesible a través de *JavaScript* e incluido en la mayor parte de los navegadores permite realizar peticiones HTTP al servidor de forma asíncrona y analizar los datos obtenidos. Aunque su nombre hace referencia al XML, en realidad este objeto permite el intercambio de cualquier tipo de datos.

En una aplicación AJAX cuando el usuario realiza una acción que requiere comunicación con el servidor, se lanza una petición asíncrona. Esta petición se lanza en segundo plano para que el usuario pueda seguir utilizando la aplicación. Al obtener la respuesta se analizan los datos recibidos para decidir que elementos de la interfaz hay que modificar. El HTML de la página siempre se modifica con *JavaScript*, nunca se recarga, por lo que se eliminan las esperas del modelo tradicional.

Como el objeto *XMLHttpRequest* también es un analizador DOM, lo normal es utilizar XML para el intercambio de datos. Este objeto también permite las transformaciones XSLT, por lo que algunas aplicaciones AJAX generan parte del HTML mediante transformaciones del XML recibido.

Lo que realmente dota de utilidad a la creación de interfaces interactivos con DHTML ha sido esta capacidad de comunicación asíncrona. El éxito de AJAX se debe en parte a la decisión de incluir el objeto *XMLHttpRequest* en los navegadores, aunque no se trata de un estándar, sino que fue inicialmente introducido por *Microsoft* en su *Internet Explorer* y posteriormente adoptado por el resto de navegadores. Sin embargo antes de la existencia de este objeto ya existía la posibilidad de una comunicación asíncrona, que en algunas aplicaciones “PRE-AJAX” ya estaba siendo utilizada. Una técnica consiste en el uso de etiquetas *IFRAME* ocultas, que permiten la carga asíncrona de un contenido que se analiza con *JavaScript* a través del DOM. Otra técnica consiste en la creación

dinámica de etiquetas *SCRIPT* que provocan la carga y ejecución asíncrona de un código *JavaScript* en el que se incluyen los datos. Este tipo de técnicas son más complicadas, pero siguen siendo válidas ya que al basarse en estándares se pueden usar en navegadores que aun no incluyen el objeto *XMLHttpRequest*.

En cuando a los datos transmitidos, otro formato muy utilizado es *JavaScript Object Notation (JSON)*. Es un formato que basado en la sintaxis *JavaScript* tiene prácticamente la misma capacidad que XML, pero con la ventaja de que ocupa mucho menos espacio y es mucho más sencillo de analizar desde *JavaScript* que un XML. Este formato también es útil para navegadores que no cuentan con un *XMLHttpRequest* con el que analizar el XML recibido.

3.1. Posibilidades de AJAX

En una aplicación AJAX el “*client engine*” asociado a toda aplicación RIA se descarga automáticamente al acceder a la pagina web de la aplicación. Se trata del *JavaScript* llamado *Motor AJAX*.

Como una aplicación AJAX no se basa en el modelo de navegación, la página web principal en la que se crea la interfaz de usuario nunca es eliminada del navegador, por lo que el código *JavaScript* y sus datos persisten durante toda la sesión del usuario. Los clientes pueden mantener datos de sesión que normalmente se sitúan en el servidor. Esto puede suponer un ahorro de recursos en los servidores, que al repartir parte de su carga entre los clientes, aumentan su capacidad para procesar mayor número de peticiones.

Por ejemplo, en una aplicación de compras, el carrito del usuario se mantiene en la sesión del servidor (datos que no se eliminan hasta después de un tiempo de inactividad elevado). Cada vez que el usuario añade un producto se realiza una petición al servidor para actualizar su sesión. En una aplicación AJAX el carrito de compra puede estar directamente en el navegador. La aplicación no necesita realizar peticiones al servidor cada vez que se añade un producto, tan solo enviar una petición final al aceptar la compra, entregando al servidor todos los datos del carrito mantenidos en el cliente.

La capacidad de comunicación asíncrona permite implementar mecanismos de *polling* eficientes, con los que el servidor puede enviar notificaciones a los clientes, o con los que se puede mantener la interfaz de usuario permanentemente actualizada de forma automática, sin necesidad de recargas de páginas HTML que provoquen un efecto no deseado. Un *polling* periódico de peticiones en AJAX permite modificar tan solo los elementos que han cambiado.

Esta característica se puede aplicar por ejemplo a aplicaciones de mensajería, para mantener los

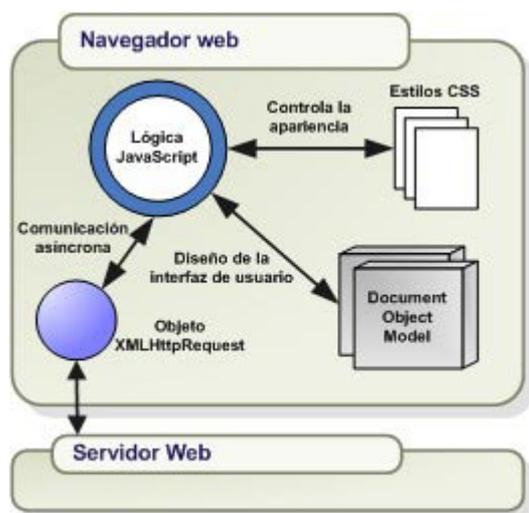


Figura 3-1: Arquitectura de las aplicaciones AJAX.

buzones de entrada y salida actualizados. O una aplicación de control de alertas, en las que éstas se recibirían automáticamente sin recargas de las páginas web del navegador.

AJAX también permite la implementación de técnicas de *prefetching* que aumentan la velocidad de respuesta de la interfaz de usuario, obteniendo por adelantado los datos que se prevé que el usuario va a necesitar.

Esta característica se está utilizando en aplicaciones como *Google Maps*. Cuando se visualiza un mapa la aplicación ya tiene descargadas porciones del mapa, ocultas alrededor del área visible. Cuando el usuario mueve el mapa con una operación de *drag and drop* las áreas nuevas se deslizan de forma inmediata y la carga de porciones se adelanta en el área oculta, aumentando el tiempo de respuesta de la interfaz.

AJAX también favorece la propagación del fenómeno conocido como "*Mashups*". Un *mashup* es una aplicación web que se compone visual y funcionalmente por trozos que provienen de distintas aplicaciones y servidores. AJAX permite que porciones de la interfaz de usuario de una aplicación sean directamente incrustadas en cualquier otra aplicación o página, y manipuladas a través de un API JavaScript.

El fenómeno de los *mashups* potencia el concepto de la cultura de la participación de Web 2.0. La colaboración en la creación de contenidos se convierte incluso en una colaboración directa en la creación de aplicaciones interactivas. Es posible crear aplicaciones nuevas basadas enteramente en la combinación de trozos de otras aplicaciones. Se introduce el concepto de la *web programable*.

El ejemplo más popular de *mashup* es el de la reutilización de *Google Maps*, que permiten mediante una API JavaScript su incrustación y personalización en cualquier aplicación. En la actualidad ya existen multitud de aplicaciones que incorporan estos mapas para aumentar su funcionalidad. Así por ejemplo muchos sitios web se dedican a combinar los servicios de *MySpace* con los de *Google Maps* para ofrecer aplicaciones interactivas en las que se muestra la ubicación geográfica de la red de usuarios o amigos de forma interactiva a través de estos mapas.

3.2. Ventajas y beneficios de AJAX

Para la Web 2.0 y en general para Internet, AJAX proporciona todos los beneficios ya comentados sobre las aplicaciones RIA, que se concretan en el concepto de *la web como plataforma*. La creación de aplicaciones completamente RIA no es posible con AJAX debido a que las tecnologías que se combinan para crear una aplicación AJAX fueron diseñadas más para la presentación de contenidos que para la

creación de aplicaciones interactivas. Sin embargo, sí proporciona una forma de acercamiento.

Lo que hace que AJAX esté imponiéndose al resto de tecnologías RIA es que su funcionamiento está basado en **tecnologías que existen de forma nativa** en los navegadores. Es como si la comunidad de desarrolladores se hubiera visto obligada a potenciar el uso del DHTML, buscando sus más oscuros entresijos para vencer la falta de cooperación que existe entre los fabricantes de navegadores para poner en común una tecnología estándar mejor adaptada al concepto de RIA.

Con una aplicación AJAX el usuario no está obligado a instalar ningún plugin en el navegador. La capacidad de "*ejecutar sin instalar*" existe con tan solo el soporte nativo del navegador (aunque es cierto que muchas de las aplicaciones AJAX más potentes requieren el uso de las versiones más avanzadas de los navegadores).

Otra ventaja importante de AJAX es el ser una **tecnología centrada en el cliente** e independiente de la tecnología y arquitectura de los servidores. Esto permite que la migración de una aplicación tradicional a AJAX se pueda realizar manteniendo intacta la mayor parte de la lógica del servidor, excepto por el hecho de que para las respuestas a las acciones del usuario se cambia la generación de HTML por el XML que el motor AJAX analiza para actualizar la interfaz de usuario.

Una forma de entender el éxito de AJAX sobre otras tecnologías mejor capacitadas para la creación de aplicaciones RIA, es observando los puntos en los que éstas fallan.

Adobe Flash Player (anteriormente *Macromedia*) podría ser la mejor opción para implementar aplicaciones RIA debido a que se calcula que el 98% de los usuarios de Internet tienen instalada alguna versión del plugin flash en sus navegadores. Incluso algunas aplicaciones de moda como *YouTube* obligan a tener el *plugin* de flash instalado para la visualización de los videos.

El problema de Flash es que es una solución completamente propietaria. El desarrollador está obligado a utilizar las herramientas de desarrollo de Adobe, así como lenguajes de programación propietarios como *ActionScript*.

Los *applets* de Java también son otra alternativa, al ser un lenguaje compilado de mayor potencia en el que se cuenta con una flexibilidad completa para la creación de interfaces de usuario. El problema de los *applets* es que el soporte ofrecido por los navegadores es muy básico, lo que obliga al usuario a instalar el JRE de Sun, que no es una práctica muy extendida.

Los controles *ActiveX* de Microsoft también proporcionan una flexibilidad incluso aun mayor, pero están limitados a su uso en el *Internet Explorer*, y presentan mayores problemas de seguridad.

Otras tecnologías como *XML-based User-interface Language (XUL)*, o *Scalable Vector Graphics (SVG)* también presentan mejores capacidades que AJAX, pero tienen problemas similares a los anteriores, o existen en un navegador en concreto o requieren la necesidad de instalación del *plugin* adecuado.

3.3. Limitaciones e inconvenientes de AJAX

En general el desarrollo de cualquier aplicación RIA siempre requerirá más esfuerzos y costes que el desarrollo de aplicaciones web tradicionales. Si las interfaces de usuario son más interactivas y funcionales es lógico que sean más complejas.

El principal problema de AJAX radica precisamente en el mismo aspecto en el que se encuentra su principal ventaja. Se apoya en tecnologías nativas del navegador, que están preparadas para el modelo tradicional de navegación por hipertexto. Crear interfaces de usuario funcionales con AJAX es complejo, deben interactuar entre sí muchas tecnologías que no fueron diseñadas para la creación de interfaces de usuario interactivos.

JavaScript es un lenguaje interpretado que tiene penalizaciones de rendimiento que no existen en los lenguajes compilados. Además es un lenguaje no tipado, lo que aumenta considerablemente el peligro de errores no detectados y hace más complicada su depuración y corrección.

El sistema de generación de interfaces mediante DHTML/CSS introduce muchas dificultades. Se trata de lenguajes declarativos basados en etiquetas, que fueron ideados para la presentación de contenidos y la navegación por hipertexto. No son lenguajes orientados a la generación de interfaces de usuarios funcionales como puedan ser las librerías *Java Swing* o las *MFC* de Microsoft. El HTML no tiene controles avanzados tales como ventanas, menús, listas, árboles, etc. Se requiere un esfuerzo adicional para simular la funcionalidad de este tipo de controles mediante unos elementos tan básicos como los del HTML.

Todas estas dificultades que presenta AJAX actualmente se están intentando acotar mediante la creación de *frameworks AJAX*, ya sea mediante APIs *JavaScript* o con la generación a través de algún lenguaje intermedio. Estos *frameworks* normalmente proporcionan controles para la interfaz de usuario con funcionalidad más avanzada y permiten desarrollar aplicaciones AJAX sin necesidad de manipulación directa del HTML. El problema de estos *frameworks* es que no existe ningún estándar común, cada fabricante ofrece su propia solución, muy diferentes unas de otras.

Como en cualquier otra aplicación RIA, en AJAX se rompe completamente con el modelo de diseño tradicional de las páginas web. En una página web tradicional los contenidos presentados al usuario pueden ser realizados fácilmente por expertos en el diseño gráfico, sin muchos conocimientos de programación, ya que utilizan un lenguaje orientado al formato. En las aplicaciones AJAX la mayor parte de la interfaz de usuario está programada. Aunque AJAX siga utilizando HTML/CSS, estos son intensamente manipulados por el *JavaScript* que crea su dinamismo. Los cambios en la apariencia de la interfaz de usuario ya no se limitan a un cambio de etiquetas, se requiere de programadores que modifiquen la lógica del *JavaScript* que crea estas interfaces.

Las tecnologías utilizadas en AJAX también tienen sus propias limitaciones, ya que muchos elementos habituales en aplicaciones nativas no pueden ser completamente simulados con DHTML, o su dificultad es enormemente elevada. Algunas características de funcionalidad de las aplicaciones nativas requieren el uso de particularidades de los navegadores, lo que aumenta el esfuerzo dedicado a mantener la compatibilidad entre todos ellos.

Otras limitaciones son las capacidades como la visualización de video o la reproducción de audio que no están disponibles en AJAX y muchas veces se acaban resolviendo con algún plugin adicional. Así es curioso encontrar algunas aplicaciones AJAX que obligan a instalar el Adobe Flash simplemente para visualizar un video, lo que es una inconsistencia ya que esto es lo que precisamente se intenta evitar con AJAX.

4. Mobile Web 2.0

El impulso para una mayor penetración de Internet en el entorno móvil debe ser enfocado directamente mediante la movilización de la Web 2.0 fija, que es ya la actualidad y el futuro de Internet.

Pero si el establecimiento de Internet en el móvil es un terreno lleno de dificultades, la movilización de Web 2.0 representa un reto aun mayor en cuanto que no existe una receta exacta que defina cuando una aplicación forma parte de la Web 2.0.

La fórmula que proponemos es aplicar el análisis detallado de cada uno de los siete principios que la configuran, para identificar la forma en la que se pueden aplicar a la Internet móvil, teniendo en cuenta las limitaciones técnicas relacionadas con la capacidad de entrada y salida de datos desde el dispositivo al usuario, es decir la potencia de las interfaces hombre-maquina de los dispositivos.

Las pantallas de los móviles son más limitadas y los teclados son insuficientes para una entrada cómoda de datos. A día de hoy el problema de la pantalla está superado en muchos terminales, que permiten la

presentación aceptable incluso de páginas HTML de la Internet fija. En la introducción de datos siguen existiendo dificultades que no superan bien ni los sistemas de pulsación sobre un teclado en pantalla o el reconocimiento de caracteres. El usuario es reacio a la introducción de datos desde el móvil por la incomodidad que representa.

Atendiendo a estas limitaciones inherentes de los dispositivos móviles, el acercamiento a Mobile Web 2.0 podría centrarse en el aspecto conceptual, encaminándolo a través de principios como el de la inteligencia colectiva. Aunque el entorno móvil no permita aun la creación de aplicaciones tan atractivas como las de la Web 2.0 fija, se pueden ir definiendo aplicaciones móviles que potencien o estén basadas en la cultura de la participación. Habilitar en las aplicaciones de la Internet móvil canales de participación.

Pero persiste la problemática de la introducción de datos. Esta cultura de participación se basa en la idea de que los usuarios usan Internet motivados ante la posibilidad de participar en la creación de contenido, y crear contenidos para aplicaciones como un *blog* o una *wiki* siempre será engorroso desde un dispositivo móvil.

Una solución a estas barreras sería el plantear los servicios de Mobile Web 2.0 como un suplemento móvil de la Web 2.0 fija. La red fija funcionaría como el centro de control desde el que los usuarios participan en la creación de contenido, dejando el entorno móvil para el consumo de esta participación o para aportaciones menores que no requieran una alta entrada de datos.

Por ejemplo la sindicación RSS en un entorno móvil permitiría seguir al tanto de lo que se produce en el entorno fijo, consumiendo lo que la Internet fija produce a través de la colaboración. También se podría asumir una participación de entrada mediante pequeños comentarios o correcciones sobre el contenido iniciado por otro usuario desde la red fija. Incluso utilizar medios del móvil como la mensajería SMS/MMS para permitir por ejemplo el envío de comentarios a una entrada de un *blog*, o la subida de imágenes/videos a servicios de compartición.

Bajo el concepto de Internet fija como centro de control de la Internet móvil se podrían ofrecer entornos con los que el usuario personaliza desde la red fija (el centro de control) las facilidades para su uso desde la red móvil.

En un escenario de este tipo el usuario podría grabar un video con el móvil y subirlo directamente a un servicio de compartición como *YouTube*. En el entorno fijo el usuario tendría libertad completa para introducir cualquier comentario y asignarle cualquier etiqueta. En un entorno móvil se podría simplificar ofreciendo una selección de etiquetas entre las que el usuario ha personalizado o entre las más usadas, lo

que hace al entorno móvil participe del concepto de "folksonomía". En este mismo escenario incluso se podrían aprovechar servicios de localización, de forma que al subir el vídeo, el móvil lo georeferenciara según la ubicación en la que ha sido grabado, y mediante un análisis de todos los datos realizar una clasificación más o menos automática.

En el plano puramente tecnológico hay que tener en cuenta que Web 2.0 es Internet y por lo tanto también lo debe de ser en el móvil. Es necesario proporcionar una Internet móvil atractiva, es decir, capacitada para ofrecer una experiencia enriquecida al usuario. Esto obliga a abandonar WAP y la creación de sitios WML, ya que suponen un marco de desarrollo muy limitado para aplicaciones interactivas (Web 2.0).

En una Mobile Web 2.0 se necesitan navegadores con las mismas capacidades que en la Internet fija. Esto es fundamental para la creación de aplicaciones RIA. Pero además se evitaría en muchos casos la necesidad de crear réplicas móviles de los sitios de Internet. Esta duplicidad de contenidos es uno de los motivos que frena una amplia generación de contenidos para el móvil. Las compañías de Internet centran la mayor parte de sus esfuerzos en la Internet fija, dejando a un lado versiones adaptadas al móvil, y más si estas tienen que duplicarse en otro lenguaje como WML.

Actualmente ya se ha empezado a realizar un acercamiento a la Web 2.0 en el entorno móvil. Un ejemplo práctico es la posibilidad que ofrece *Google* de acceder a sus servicios *Google Maps* o *GMail* desde el móvil (Fig. 4-1).



Figura 4-1: Imagen del servicio *Google Maps* accesible desde el móvil a través de la descarga e instalación de una aplicación cliente nativa.

Pero precisamente este ejemplo permite observar los impedimentos que plantean las barreras tecnológicas del móvil. El servicio *Google Maps* para el móvil no se utiliza desde un navegador como en la Internet fija. El usuario no ejecuta una aplicación directamente en Internet, necesita el paso previo de descarga e instalación de un pequeño cliente específico y compatible para su terminal.

Esto presenta una incongruencia significativa. Se tienen terminales con la tecnología necesaria para mostrar los mapas de *Google* con todo detalle, pero que no son capaces de mostrarlos a través del navegador. Esto podría interpretarse como que la disponibilidad de AJAX en los móviles más que una cuestión de limitaciones técnicas es una falta de compromiso por parte de fabricantes de introducir navegadores completamente funcionales.

5. AJAX Mobile

La reflexión anterior es la que nos lleva a considerar que AJAX sería la mejor forma de acercar la Web 2.0 al móvil desde el plano tecnológico. No sería necesaria la introducción de tecnologías especializadas, si no tan solo la introducción de navegadores que soporten los mismos estándares que los de la Internet fija. En un móvil con la tecnología suficiente para ejecutar la aplicación *Google Maps* no debería ser un impedimento la inclusión de un navegador completo que permitiera aplicar el concepto de la web como plataforma en el móvil.

Al igual que en la Internet fija AJAX ha sido la tecnología que ha permitido extender la creación de interfaces de usuario enriquecidas para la Web 2.0, en el entorno móvil AJAX también podría jugar el mismo papel, y de esta forma aplicar el principio de *la web como plataforma* también en la Internet móvil.

En la Internet fija la falta de una tecnología estándar para la creación de aplicaciones RIA que sea aceptada por todos los navegadores ha sido la principal razón de la adopción de AJAX. El panorama actual de la Internet móvil predice una situación similar a la de la Internet fija.

Actualmente ya están apareciendo versiones para móvil de tecnologías RIA como Flash o como SVG. Pero sigue sin llegarse a un acuerdo para que todos los navegadores incluyan una misma tecnología. Por otro lado, el obligar a que todos los fabricantes de móviles incluyan una tecnología propietaria como Adobe Flash sería como obligar a que todos los móviles llevaran un mismo navegador como Internet Explorer.

Por tanto la adopción de AJAX en el entorno móvil se presenta como una opción a considerar para llevar las aplicaciones RIA a una Mobile Web 2.0.

Conforme aumenta la potencia de los dispositivos móviles aumenta la tendencia a incluir navegadores más completos, y por tanto cada vez mejor preparados para AJAX. Compatibilizar un navegador con AJAX tan solo supone que soporte de forma completa las tecnologías habituales de Internet como HTML, CSS, DOM, *JavaScript* y el objeto *XMLHttpRequest*. Es más probable que los fabricantes de navegadores para móviles empiecen a replicar de forma completa las tecnologías nativas de la Internet fija como el DHTML y el *JavaScript* (que ya existen), que el ponerse de acuerdo para introducir alguna otra tecnología no relacionada de forma tan directa con la navegación por páginas web (novedosa e inexistente). En esta línea, el navegador de Opera para teléfonos móviles utilizado por la mayoría de los últimos modelos lanzados por Nokia ya ha anunciado su compatibilidad con AJAX.

Sería suficiente con implementar las anteriores tecnologías ajustándose a las especificaciones de los organismos de estandarización. Incluso el objeto no estándar *XMLHttpRequest* sería opcional, ya que con DHTML se pueden utilizar otras técnicas para la comunicación asíncrona, como ya se ha expuesto con anterioridad. Aunque la adopción de estándares “*de facto*” como éste, facilitaría la transición de AJAX hacia el móvil.

Actualmente ya existen dispositivos móviles con navegadores capaces de visualizar correctamente muchas páginas web de la Internet fija, pero al no incluir de forma completa sus tecnologías, la mayoría no permite la ejecución de las aplicaciones AJAX de la Internet fija.

Lo que hay que descartar completamente es un soporte de AJAX basado en la instalación de algún tipo de plugin, ya que es precisamente esto lo que se intenta evitar utilizando AJAX. En resumidas cuentas AJAX en el móvil significa apostar por navegadores con los estándares de Internet implementados al completo.

5.1. Beneficios de AJAX Mobile

En general la introducción de AJAX en el entorno móvil proporcionará los mismos beneficios que ya han sido expuestos para la Internet fija. En resumen, se trata de la posibilidad de **convertir Internet móvil en una plataforma de ejecución**, mediante la creación de aplicaciones RIA, o al menos mediante la creación de aplicaciones web accesibles en movilidad y con interfaces de usuario más interactivas y amigables que atraigan a los usuarios y vitalicen la Internet móvil.

En el entorno móvil también existen problemas para la distribución de aplicaciones nativas. Los dispositivos móviles cuentan con diferentes sistemas operativos o plataformas nativas de ejecución, incompatibles entre ellas. Esto obliga a que los desarrolladores inviertan esfuerzos adicionales en

preparar diferentes distribuciones de la misma aplicación, lo que repercute en un aumento de los costes de producción. Incluso las aplicaciones para J2ME que deberían ser universales, al final necesitan de adaptaciones, ya sea por diferencias en las librerías incluidas o por particularidades de los dispositivos.

Algunas de estas aplicaciones nativas se podrían convertir en aplicaciones AJAX, lo que redundaría en la **compatibilidad** y reduciría el número de distribuciones a desarrollar. Además al ejecutarse directamente desde la web, se elimina la necesidad de instalación, lo que permite al usuario **ahorrar memoria**, que en estos casos es limitada.

Otra ventaja que aportaría el uso de AJAX al entorno móvil es su **economía en la transmisión de datos a través de la red**. En un entorno con un ancho de banda más limitado esto representa una ventaja adicional. En una aplicación AJAX se produce una mayor descarga de datos al iniciar la aplicación, debido a la necesidad de descargar el *Motor AJAX*. Pero durante el uso de la aplicación, el tráfico por la red es mucho menos voluminoso, ya que una aplicación AJAX solo transmite datos, y una aplicación web tradicional produce descargas de elementos de presentación durante la navegación, que son mucho más voluminosos.

5.2. Limitaciones de AJAX Mobile

La adopción de AJAX en el entorno móvil hereda también los mismos problemas que tienen las aplicaciones web tradicionales. Es decir, las limitaciones de las interfaces hombre-máquina inherentes a cualquier dispositivo móvil.

Un problema adicional de las aplicaciones AJAX en el móvil, es que su mayor complejidad provoca que se consuman más recursos en la parte cliente. En los dispositivos móviles los recursos están más limitados, contando con una menor potencia computacional y memorias más reducidas.

No obstante esta limitación no es tan problemática, ya que AJAX reparte recursos entre el cliente y el servidor. Actualmente los dispositivos móviles ya utilizan muchas aplicaciones instalables que se ejecutan íntegramente utilizando sus recursos de computación de forma intensa. Por lo que en un dispositivo con suficiente potencia para ejecutar por ejemplo aplicaciones J2ME no debería representar mucho problema la ejecución de aplicaciones AJAX, excepto quizás por la penalización que provoca el uso de un lenguaje de programación interpretado como *JavaScript*.

Las aplicaciones AJAX para un entorno móvil también deberían tener en cuenta que en este entorno los problemas de pérdida de conexión son más frecuentes, y quizás sería necesario hacer que las

aplicaciones fueran más robustas y estuvieran preparadas para estas pérdidas de conexión.

Aunque actualmente existen dispositivos móviles que soportan AJAX, es importante que esta práctica se generalice para superar la alta fragmentación del mercado de terminales (la compatibilidad de, por ejemplo aplicaciones J2ME es una entelequia) genera una serie de barreras que impediría reproducir el éxito que las aplicaciones Web 2.0 basadas en AJAX han tenido en la Internet fija.

6. Conclusiones

AJAX es la solución elegida de facto por la comunidad de desarrolladores en Internet para ofrecer interfaces avanzadas, con una apariencia lo más amigable posible e independiente (hasta cierto punto) de los navegadores, al basarse en tecnologías nativas incorporadas por todos los fabricantes. La pregunta es “¿Es replicable el modelo en el mundo móvil?”. Según algunos expertos, sí le auguran posibilidades de éxito, siempre sin desmerecer otras tecnologías cliente como J2ME, *FlashLite* o *RichMedia*, pero para ello hay que alcanzar un parque de terminales cuyos navegadores soporten tecnologías básicas en el mundo fijo como *JavaScript*. Cualquier otra cosa no sería AJAX.

Mientras tanto, se pueden recopilar argumentos a favor o en contra de AJAX como paradigma para vitalizar el negocio de Internet Móvil, pero no serán más que intentos de trasladar al mundo móvil los conceptos previamente discutidos a favor y en contra de AJAX como tecnología bandera de la Web 2.0.

Agradecimientos

A todos nuestros compañeros de Telefónica I+D, con los que compartimos el trabajo diario y que de una forma directa o indirecta han aportado valiosas ideas para la confección de este artículo.

Bibliografía

- Open Gardens, Mobile Web 2.0. Disponible en <http://www.opengardens.net>
- Opera Platform. AJAX applications on mobile phones. Disponible en <http://www.opera.com/products/mobile/platform>
- O'Reilly Media, “*What is Web 2.0*”. Disponible en <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>
- Web 2.0 Conference. Disponible en <http://www.web2con.com>
- Dave Crane, Eric Pascarella y Darren James, *Ajax in Action*, Manning Publications, Greenwich, 2006.
- The World Wide Web Consortium (W3C). Disponible en <http://www.w3.org>