

AJAX: un análisis tecnológico y posibilidades metodológicas

Autor

Patricia Bazán

pbaz@ada.info.unlp.edu.ar

LINTI

Laboratorio de Investigación en Nuevas Tecnologías Informáticas

Facultad de Informática – UNLP

Universidad Nacional de La Plata

Director: Lic. Francisco Javier Díaz

Calle 50 y 115 – 1er Piso – la Plata (1900) – Buenos Aires – Argentina

fax: 0221-423-0124 – tel: 0221-422-3528

1. Introducción

Este trabajo presenta un análisis desde el punto de vista tecnológico de AJAX y su la posibilidad de alcanzar la inserción de dicha tecnología en un marco conceptual y metodológico para el desarrollo de software.

El objetivo del trabajo es detallar conceptos y tecnologías subyacentes, encontrar sus fundamentos y plantear un uso adecuado de los mismos en la construcción de software robusto, tecnológicamente moderno y metodológicamente correcto.

2. Conceptos y definiciones

AJAX es una etiqueta que refiere un repertorio de técnicas basadas en los navegadores para implementar aplicaciones Web en las cuales los elementos y contenido de las páginas son recuperados de manera asincrónica en *background*, y por la cual el despliegue de la página se actualiza de manera incremental sin necesidad de desplegar la página completa.

Lo nuevo en la evolución actual en cuanto a implementaciones de navegadores y capacidad de procesamiento del hardware es que permite que gran cantidad de código JavaScript se ejecute correcta y consistentemente con robustez y rendimiento aceptables.

Es el asíncrono de JavaScript + XML para un modelo de desarrollo basado en estándares para RIA (Rich Internet Applications). AJAX difiere en tres aspectos de las formas clásicas de aplicaciones Web basadas en páginas. Estos son:

1. una máquina del lado del cliente actúa como intermediario entre la UI y el servidor
2. la máquina del lado del cliente maneja la actividad del usuario en lugar de enviar el requerimiento al servidor.
3. los datos XML son transferidos entre la máquina del lado del cliente y el servidor, permitiendo el refresco de páginas parciales en lugar de páginas completas.

AJAX no es un lenguaje de programación. Es una técnica para crear mejores y más rápidas aplicaciones Web usando sus mismas tecnologías (HTML, CSS, JavaScript y XML). Dado que usa funcionalidad nativa de los navegadores no requiere plug-in. Usa el objeto de los navegadores XMLHttpRequest, que permite hacer requerimientos al servidor sin refresco completo de la página.

3. Aspecto tecnológico

Dado que AJAX es un conjunto de tecnologías es posible desarrollar aplicaciones AJAX desde cero sin usar ninguna herramienta de desarrollo. Sin embargo, existen bloques constructores AJAX que aceleran el proceso de desarrollo para mediana a gran escala.

Estos bloques constructores se clasifican en cuatro categorías:

- ❖ **Snippets:** pequeños fragmentos de código. Son incorporados sin grandes cambios a la arquitectura de la aplicación.
- ❖ **Widgets:** componentes de UI autocontenidas. Incorpora nuevos elementos de interface (no sólo extiende las existentes). Ej.: un calendario, un menú jerárquico, un menú acordeón.
- ❖ **Frameworks:** un entorno de ejecución del lado del cliente con utilidad de funciones y *widgets*. En este caso, el desarrollador debe atravesar el código existente y re-escribirlo en el nuevo *front-end*.
- ❖ **Frameworks avanzados:** un *framework* con herramientas de desarrollo y componentes del lado del servidor que también tiene componentes del lado del cliente. Proveen soporte para SOA, repositorio de datos y de metadatos, soporte para desarrollo en equipo y gestión de aplicación.

	Construir /Comprar	Riesgo técnico	Riesgo de negocio	Beneficio de usuario	Beneficio de negocio
Snippets	Construir	Bajo	Bajo	Inmediato	Bajo
Widgets	Construir	Bajo	Bajo	Inmediato	Bajo
Framework del lado del cliente	Comprar	Medio	Medio	Mediano plazo	Medio
Framework del lado del cliente y del servidor	Comprar	Alto	Medio	Largo plazo	Alto

Tabla 1 – Características en cada bloque constructor AJAX

4. Posibilidades metodológicas

Dados los beneficios enunciados, ¿cuáles serían los motivos para no adoptar ampliamente AJAX?

- **Proyectos, widgets y tollkits AJAX sin soporte:** existen varios proyectos *open source* de AJAX pero son soluciones fragmentadas sin soporte de una compañía.
- **Carencia de buenas prácticas establecidas:** sin ellas, la técnica de prueba y error provoca demoras y pérdida de tiempo.
- **Pobre integración al ciclo de vida del software:** no hay herramientas para acelerar el testeo y el *debugging* y para integrar AJAX a los procesos de desarrollo existente, por lo tanto, prolifera el código “spaguetti”
- **Carencia de soporte externo y entrenamiento:** producto se ser una tecnología no soportada comercialmente.
- **Riesgo de violación de propiedad intelectual:** los proyectos AJAX incluyen código fuente disponible en Internet.

Web 2.0 es un término popular que describe una nueva generación de aplicaciones Web altamente interactivas. Esta interacción debe ser resuelta por un nuevo paradigma de construcción interfase de usuario conocida como RIA (Rich Internet Applications).

Web 2.0 tiene el desafío de no ser solamente una tecnología sino también un conjunto de atributos con dimensión social que definen nuevos modelos de negocio, nuevos patrones de diseño mas orientados al usuario y un paradigma de construcción de software mas descentralizado.

La tecnología RIA es el pegamento que enlaza la riqueza de las aplicaciones Web con la riqueza de las aplicaciones de escritorio. Su potencia radica en trasladar el procesamiento de la interfase de usuario del servidor al cliente, manteniendo el acceso a los datos en la aplicación servidora. Varias tecnologías RIA se utilizan hoy siendo AJAX la más ampliamente usada.

	Tecnologías RIA			
	AJAX <i>Open Source</i>	AJAX Comercial	Java del lado del cliente	Flash
Soporte Web estándar	Excelente	Excelente	Regular	Mediocre
Soporte del vendedor	Pobre	Excelente	Mediocre	Excelente
Tamaño de download	Excelente	Excelente	Pobre	Regular
Instalación cero sin plug-in	SI	SI	NO	NO
Performance	Regular a mala	Excelente	Regular	Excelente
Facilidad de desarrollo	Regular a mala	Excelente	Regular	Excelente
Madurez	Mediocre	Buena	Excelente	buena

Tabla 2 – Comparación de características entre las distintas tecnologías RIA

AJAX es un subconjunto de la amplia categoría de tecnología RIA. Ambas tienen cuestiones en común: cubren el amplio espectro que va de aplicaciones *thin-Web* con HTML puro hasta las aplicaciones cliente/servidor tradicional y tratan de evitar los defectos de ambas.

RIA y AJAX emulan el cliente/servidor tradicional pero con un mejor “*deployment*”, amplia gama de plataformas de ejecución (a través de máquinas virtuales y ambientes de ejecución multiplataforma) y mejor calidad en cuanto a seguridad.

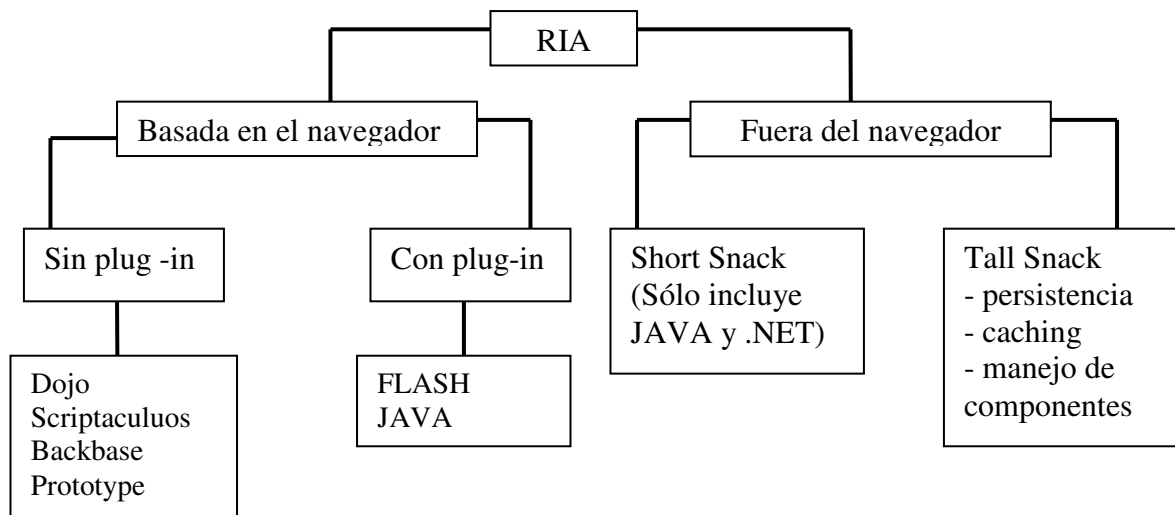


Figura 1 – Visión jerárquica de AJAX y RIA

La tecnología AJAX amplia y madura apunta a un gran conjunto de desarrolladores, más allá de los expertos en tecnologías del lado del cliente como CSS, DOM y JavaScript.

5. Inserción en las empresas

Existe un gran grupo de desarrolladores que construyen aplicaciones AJAX-enablement con plataformas del lado del servidor como JSEE o ASP.NET o frameworks específicos como Struts, Spring o JSF.

Por lo tanto, la taxonomía en tiempo de diseño se transforma en algo tan importante como la definición de los entornos de ejecución.

Lo que en verdad es importante en ese contexto es si la plataforma RIA soporta un lenguaje familiar del lado del servidor, se conecta con sistemas SOA y si se integra bien con herramientas de desarrollo del lado del servidor como Eclipse o Visual Studio.

Esto representa una divergencia entre lo que se denomina “mercado global” o cara externa de los desarrollos Web como Yahoo o Google, del “mercado particular” o cara interna de los desarrollos de negocio de tipo corporativo. Los primeros son claramente centrados en Web y frecuentemente se autoabastecen (no requiere soporte técnico externo ni proveedores de sofá) y están abocados a la introducción de nuevas tecnologías. Los segundos tienen otra realidad, deben sostener en producción las denominadas aplicaciones “*legacy*” y poseen un gran bagaje de tareas relacionadas para dichas aplicaciones, por ende, suelen estar cautivos de los productos por sobre la tecnología cruda.

Desde la perspectiva del desarrollador la tecnología AJAX puede dividirse en tres:

- Centrada en el cliente: herramientas y framework centradas en Web que requieren que el desarrollador escriba código JavaScript.
- Centrado en el servidor: herramientas front-end que extienden los frameworks para programar del lado del servidor.
- Ni del lado del cliente ni del lado del servidor. Los desarrolladores diseñan la UI y el comportamiento usando una descripción no procedural basada en XML.

Los beneficios del XML declarativo es que hace más fácil manejar la UI para un portfolio de aplicaciones que comparten el mismo diseño.

La desventaja es que no existen estándares, son propietarios de cada framework.

6. Limitaciones de AJAX

Una metáfora que caracteriza la diferencia entre la tecnología RIA basada en AJAX de otras tecnologías más exhaustivas o robustas, es la metáfora “del cielo y la tierra”.

Microsoft introdujo esta metáfora para instruir a sus desarrolladores acerca de la diferencia entre AJAX y lo que ellos denominan “*smart client*”, que sustenta el subsistema grafico de Windows Vista.

Se puede decir que AJAX ha “emergido de la tierra” permitiendo nuevos estilos de interacción a través de refresco parcial y requerimientos asincrónicos. El “cielo” se ajusta a aplicaciones que requieren una interacción mas estrecha con el usuario mas allá de los confines del navegador. La diferencia sustancial esta dada por la necesidad de ejecutar operaciones en modo desconectado.

Los detractores de AJAX identifican a esta última como la característica “asesina” de la tecnología RIA, pero pierden de vista, que aún resuelto este aspecto por parte de AJAX, existen otras limitaciones como la integración limitada al cliente solamente, la seguridad (los nuevos patrones de comunicación pueden introducir nuevos riesgos de seguridad) y la mantenibilidad del código que para muchos puede resultar poco familiar o raro.

7. Riesgos y seguridades al adoptar AJAX

La incorporación de una nueva tecnología siempre trae aparejado riesgos y seguridades. Lo importante es tener presente cada una para minimizar las primeras y potenciar las segundas. En el caso de AJAX en particular, y habida cuenta que se ha definido como un conjunto de tecnologías que pueden

converger en un verdadero cambio de punto de vista en el desarrollo de software, es que se hace aun más necesario evaluar estos dos aspectos.

Entre las seguridades con las que se cuentan están;

- Es fácil comenzar en AJAX con *snippets* (denominado Nivel 1 de AJAX) y *widgets* (denominado nivel 2 de AJAX).
- Adopción de un *framework* robusto más que desarrollar uno propio (nivel 3 y 4 de AJAX).
- Fomentar el procesamiento del lado del servidor.
- Incorporación de diseño centrado en la usabilidad a los procesos de negocio, independientemente de AJAX.

Dentro de los riesgos que se corren utilizando AJAX están:

- Es fácil comenzar en AJAX con *snippets* y *widgets* y detenerse allí.
- Adopción de un *framework* robusto más que desarrollar uno propio (nivel 3 y 4 de AJAX) y que el proveedor desaparezca del mercado.
- Enfocarse solo en la tecnología más que en el problema y sus objetivos.
- Tener la visión que solo es un proyecto del lado del cliente.
- Ignorar las implicancias de seguridad que se presentan en esta interacción de grano fino entre cliente y servidor.

8. Conclusiones

AJAX como tecnología tiene varias ventajas, como el estar basada en estándares, no usar plug-ins, ajustarse al flujo de código habitual, no requerir más conocimientos que JavaScript y trabajar en todos los lenguajes de programación y todas las plataformas.

Desde lo metodológico, hay variantes para adoptar esta tecnología que van desde la simple inclusión de *snippets* hasta construir desarrollos de software completos con entornos de desarrollo basados en tecnología RIA.

Esta gama de variantes deberá ser evaluada por los responsables tecnológicos de los proyectos informáticos a la hora de decidir desarrollar aplicaciones cuyo modelo de interacción, modelo de negocios, seguridad, mantenibilidad y demás requisitos exigidos por el mercado para los desarrollos modernos, se encuentren a la altura de las circunstancias.

9. Referencias

1. Phifer G. 2007. “*AJAX Limits, RIA Technology Risks and User Experience Possibilities*” Gartner Application Integration & Web Services Summit. - Backbone Client Edition 4.0.2. Beginner Tutorial.
2. Backbone Client Edition 4.0.2. Beginner Tutorial.
3. Backbone Client Edition 4.0.2. Technical Overview
4. Bazan 2008. “*Conceptos y generalidades sobre AJAX y tecnología RIA*”. Informe técnico interno elaborado en el LINTI. Facultad de Informática. UNLP.