

6.1 INTRODUCCIÓN

Es importante analizar los dos grandes grupos en el desarrollo de software, que son el software propietario y el software libre.

El software propietario también llamado software no libre o privado. Se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o que su código fuente no está disponible o el acceso a éste se encuentra restringido.

El software libre también llamado de código abierto, es aquel que puede ser distribuido, modificado, copiado y usado; por lo tanto, se tiene el código fuente a nuestra disposición. Dentro de software libre hay que tomar en cuenta ciertas consideraciones:

- El software de dominio público significa que no está protegido por el copyright, por lo tanto, podrían generarse versiones no libres del mismo.
- El software libre protegido con copyleft impide a los redistribuidores incluir algún tipo de restricción.

Conocidos estos conceptos se define que Ajax representa al software libre y que Adobe Flex al software propietario, pero las dos tecnologías nos permiten construir aplicaciones visual y funcionalmente similares, por tal razón es importante realizar un análisis comparativo entre estas, determinando las características, ventajas y desventajas, a través de elementos de evaluación los cuales aseguren obtener conclusiones teóricas y prácticas que avalicen el estudio de esta investigación.

También es importante mencionar que se adoptara los criterios más importantes de evaluación de calidad de software según la norma ISO 9126 señalada en el Anexo 2.2, estos criterios serán:

- Funcionalidad: Integración con sistemas existentes como base de datos, sistemas operativos y servicios Web.

- Eficiencia: En la utilización de recursos ya sean en hardware, software entre otros.
- Portabilidad: En cuanto a compatibilidad con diferentes plataformas y arquitecturas.

6.2 ELEMENTOS PARA LA EVALUACIÓN

Para el análisis comparativo se identifican a continuación los elementos que serán motivo de nuestro estudio:

1. Entornos de desarrollo
2. Lenguajes
3. Arquitectura
4. Integridad tecnológica
5. Costos en infraestructura tecnológica
6. Estándares
7. Seguridad
8. Otros:
 - Amplitud de uso en el desarrollo de software
 - Facilidad de uso y programación

Cabe recalcar que los elementos nombrados anteriormente, serán examinados en dos secciones: La primera contempla el análisis teórico que consiste en determinar la mejor tecnología en base a los argumentos expuestos ya en los capítulos IV y V.

La segunda sección contempla el análisis práctico que consiste en comprobar que tecnología es la mejor a través de un sencillo aplicativo (prototipo), además se detallará el proceso de instalación, desarrollo y publicación de un formulario de autenticación de usuarios, una vez terminado se someterá a una prueba de usabilidad determinando si se cumplen los factores más importantes de la misma.

6.3 ANÁLISIS COMPARATIVO TEÓRICO

1. ENTORNOS DE DESARROLLO (IDE)

Adobe ofrece la posibilidad de poder descargar una versión de prueba de Flex con su IDE Flex Builder 2.0. Este editor trabaja con MXML, esta integrado con un depurador de código que detecta errores en la sintaxis del lenguaje MXML, pero no puede detectar errores en el código ActionScript o componentes derivados, las advertencias de los errores no son lo suficientemente claras para identificar y corregir el problema de la aplicación. Dispone también de un excelente pre-visualizador, para poder ver el aspecto que tendrá nuestra aplicación al ejecutarla en el navegador.

Ajax, al ser un conjunto de varias tecnologías se adapta a diferentes IDE's siempre que soporten HTML/XHTML, JavaScript, XML u otro lenguaje de marcado pero si no se tiene un IDE adecuado se puede utilizar Bloc de notas o editores **WYSIWYG**. En este punto es importante recalcar que existen IDE's de código abierto similares al entorno Eclipse como Aptana el cual nos servirá para la construcción del código.

El desarrollo de código JavaScript no tiene un IDE para depurarlo sin embargo existen mecanismos que permiten corregir los errores que se presente por ejemplo usar alertas en el código (alert) o depurar la aplicación en los depuradores de los navegadores como la herramienta FireBug de Mozilla Firefox, como se puede observar en la siguiente figura:

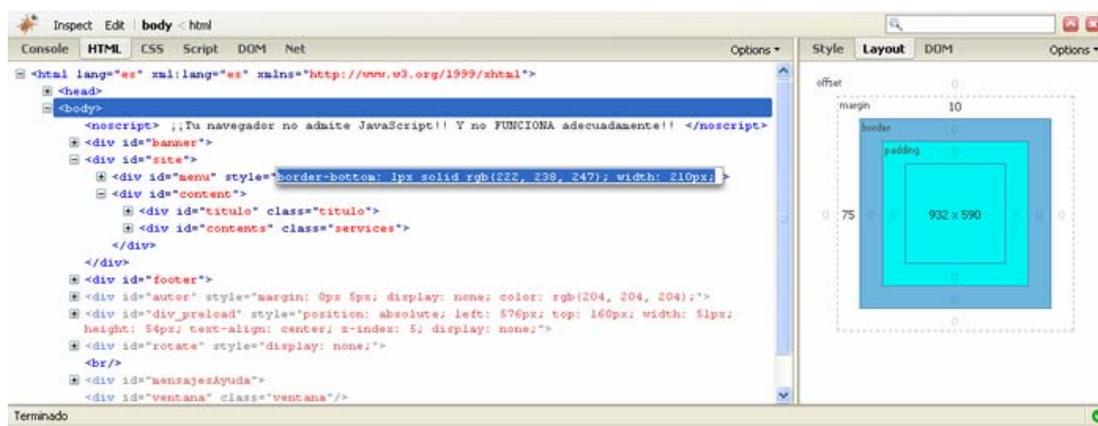


Figura 6.1 Herramienta de depuración de aplicaciones AJAX

2. LENGUAJES

Lenguajes de marcado

Los lenguajes de marcado de ambas tecnologías se basan en los mismos principios y guardan ciertas similitudes, se basan en XML y utilizan un lenguaje de script estandarizado para añadir interacción a las aplicaciones.

En el caso de Flex, el lenguaje de marcado es MXML y utiliza ActionScript en sus versiones 2.0 y 3.0 como lenguaje de script.

En Ajax, como lenguaje de marcado XML y JavaScript como lenguaje de script. Pero la gran flexibilidad de Ajax nos permite trabajar con lenguajes de marcado derivados y notaciones como JSON.

Estilos

Flex permite la utilización de archivos CSS para añadir estilo a los componentes, esta característica es importante para administrar el aspecto visual de las aplicaciones tanto a nivel de controles, gráficas y eventos.

Ajax también permite administrar la apariencia de los componentes de la interfaz desde el texto hasta los controles de formularios mediante archivos CSS para darle estilo a la aplicación. Y si nos gusta trabajar con XML podemos usar plantillas de diseño XSLT.

3. ARQUITECTURAS

Optimización de las aplicaciones

Flex permite utilizar Runtime Shared Libraries (bibliotecas compartidas en tiempo de ejecución) donde las aplicaciones comparten funciones comunes. Con este sistema se reduce el tamaño de los archivos .swf ya que se evita la duplicación de código.

Flex también dispone del ActionScript Profiler, que prepara el código ActionScript para que su interpretación en Flash Player, sea más rápida.

Adobe incorporó últimamente Flex-Ajax Bridge (FABridge) una biblioteca que permite realizar la comunicación bidireccional con el servidor a través de JavaScript

Ajax al emplear una comunicación asincrónica permite una descarga progresiva de la aplicación, según la necesidad del usuario. De esta manera se canaliza el ancho de banda utilizado al abrir una URL de la aplicación por efecto se esta reduciendo el trafico de red entre el cliente y el servidor. Para reducir el tiempo del primer acceso del cliente se optimiza el diseño con técnicas CSS y usando la reutilización de archivos externos JavaScript.

Aspectos Gráficos

Flex Charting es una biblioteca con una gran variedad de gráficos estadísticos que permiten personalizar y poner efectos interactivos a los gráficos, también tienen la posibilidad de ofrecer datos gráficos en tiempo real.

Ajax permite construir gráficos mediante **SVG**, librerías JavaScript que permiten desplegar diagramas estadísticos apoyados en gráficos vectoriales.

4. INTEGRIDAD TECNOLÓGICA

De terceros

Flex permite trabajar con tecnologías en ambientes distribuidos como Java y .Net así como también con lenguajes de servidor como JSP, PHP, ASP.

Ajax por su parte tiene integridad tecnológica con todos los lenguajes de servidor y tecnologías que funcionen bajo un navegador moderno. Adicionalmente permite trabajar con la incorporación de nuevas tecnologías como: Ruby onRails.

Base de Datos

Tanto Ajax como Flex al pertenecer a la capa de presentación son totalmente independientes de cualquier Base de Datos, por esta razón depende mucho de la lógica comercial de nuestra aplicación pero podemos asegurarnos que tenemos un

amplio campo de integración con base de datos como: MySQL, SQL, Postgress, Informix, Sybase, Oracle, entre muchas otras.

Componentes de integración de aplicaciones

Como hemos ya analizado Ajax y Flex nos permiten tener una gran integridad con muchas arquitecturas de aplicaciones Web y más con los componentes para llamar librerías externas o controladores internos de la aplicación. Mencionamos a continuación los servidores más importantes:

- Apache Tomcat 5.5.x
- BEA WebLogic 8.1 o superior
- IBM WebSphere 5.1.1.x o superior
- JBoss 4.0.3 SP1 and 4.0.4
- Oracle AS 10G (10.1.3)

Sistemas Operativos y Navegadores

Es sumamente importante analizar la diversidad de sistemas operativos donde se puede usar una aplicación Web por esto Ajax como Flex garantizan su funcionalidad de IDEs en varios sistemas operativos como:

- Microsoft Windows: XP Profesional, 2000 Server, 2003 Server (32 o 64 bits).
- Linux: Red Hat Advanced Server 3.x, SUSE Enterprise Server 9.
- Solaris SPARC 9.

También es importante donde trabajan nuestras aplicaciones es decir los navegadores, tenemos los siguientes:

- Microsoft Internet Explorer 5.x o superiores.
- Mozilla Firefox 1.x o superiores.
- Netscape 7.x o superiores.
- Opera 7.x o superiores.

Servicios Web

Las dos tecnologías permiten el uso de servicios Web, mediante SOAP o RPC. Y en los dos casos, el uso de estos servicios se realiza mediante la utilización de un proxy que actúa de puente entre las aplicaciones y el servicio.

5. COSTOS EN INFRAESTRUCTURA TECNOLÓGICA

- **Costo**

Flex 2 es una solución de desarrollo enfocado al desarrollo de aplicaciones de mayor impacto para grandes empresas. Así lo demuestra, su costo que supone una licencia de 1 máquina para el servidor de presentación que es de alrededor de 10.000 dólares. Esto supone un gran inconveniente para las PYMES porque es muy elevado y se puede realizar las mismas aplicaciones con otras tecnologías, como Flash y JSP, aunque con más dificultad. Por otro lado el servicio de hosting para alojamiento de aplicaciones Flex, es difícil de conseguir y se ofrecen a precios elevados. Para este caso se podría contratar un servicio de hosting dedicado (lo cual es costoso).

En cambio, para el desarrollo aplicaciones Ajax existen dos tendencias la primera no cuesta dinero, ni depende de una licencia comercial y la segunda el software propietario, empresas que han implementado a sus frameworks o arquitecturas componentes para soluciones basadas en AJAX, estas licencias comerciales van desde los 2.000 dólares.

- **Soporte de ayuda**

En el sitio Web de adobe (www.adobe.com) se pueden visualizar e incluso descargar algunos ejemplos de aplicaciones Flex, ayudándonos a los desarrolladores a programar funcionalidades para estas aplicaciones. También existen blogs donde se publican códigos fuente. En el mercado encontramos libros que también sirven de ayuda. Al tener una licencia comercial del producto se puede acceder a la ayuda de expertos sobre Flex.

Para Ajax se encuentran manuales, tutoriales, documentos técnicos, ejemplos, y visualización de aplicaciones reales en varios sitios de la Web. Además, existen numerosos blogs especializados en Ajax, libros publicados de consulta y todo el material de soporte en formato digital.

- **Licencias**

Adobe ofrece una licencia sin coste alguno para uso no comercial de sus aplicaciones, esta nos sirve para estudiar dicha tecnología por 30 días. También tiene varios tipos de licencias como: El Programa por volumen de compras de licencias de Macromedia (MVLP).

Para Ajax tenemos dos opciones, licencias GNU GPL (General Public License) que está diseñada para garantizar la libertad de compartir y modificar el software. Pero por otro lado existe licencias corporativas que cada empresa restringe las condiciones del software.

6. ESTÁNDARES

En los capítulos IV y V se estudió Ajax como Flex respectivamente y podemos definir que las dos tecnologías se basan en los siguientes estándares: SOAP, HTTP, HTTPS, XML, DOM, CSS, XHTML, Ecma-262, SVG.

7. SEGURIDAD

Ninguna aplicación que funcione sobre Internet esta exenta de ataques y vulnerabilidades es por eso que tanto Flex con su robusta arquitectura, como Ajax tiene medidas de seguridad estas fueron descritas en los capítulos IV y V.

8. OTROS

- **Amplitud de uso en el desarrollo de software**

Flex al tener un alto costo en sus licencias comerciales, el uso en el desarrollo de software es limitado a empresas de mayor capital económico.

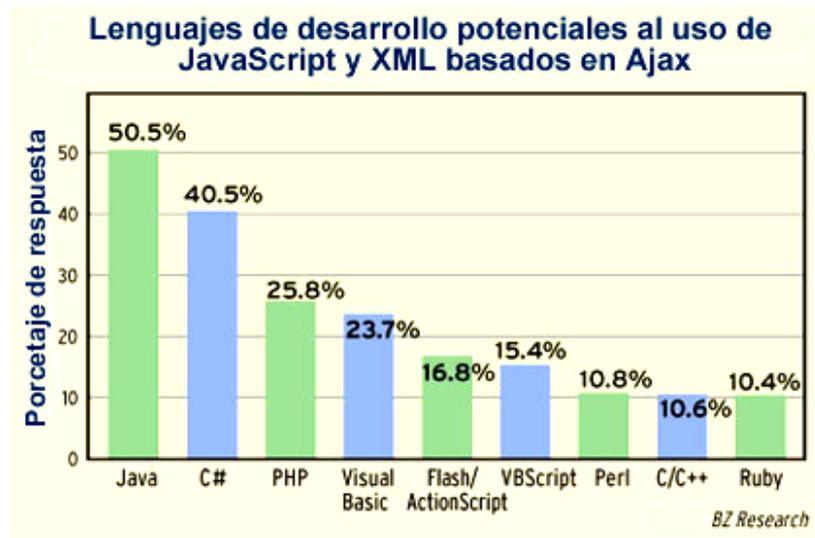


Figura 6.2 Tendencias para el desarrollo basado en AJAX

El uso de Ajax está creciendo más allá de lo esperado y está tomando sostenimiento en el mundo del desarrollo de software, ya que estudios estadísticos realizados confirman este hecho como el de BZ Research como se muestra en la figura 6.1.

- **Facilidad de uso y programación**

Flex al tener un poderoso IDE el desarrollo de código es más fácil e intuitivo, también su código no es nada complejo ya que se basa en XML, lo más complicado es usar ActionScript 3.0 para darle funcionalidad al manejo de eventos ya que exige tener bastante experiencia en este campo.

Ajax al ser un conjunto de tecnologías por ende exige sólidos conocimientos para realizar aplicaciones altamente interactivas. Lo más complejo es unir todas las tecnologías en un solo objetivo.

A continuación se presenta una tabla resumen de la comparativa teórica entre Flex y Ajax:

Elemento	FLEX	AJAX
Entornos de desarrollo		
IDE	Bloc de notas Flex Builder 2.0	Bloc de notas IDE de código abierto por ejemplo Aptana 0.27
Depuradores de código	Flex Builder incluye un depurador pero los errores son complejos de resolver cuando se incorpora objetos actionScript	Los IDE's para Ajax no tiene depuradores, pero al ser JavaScript podemos usar los depuradores de los navegadores por ejemplo: FireBug de Firebox
Lenguajes		
Lenguaje de script	AS 2.0, 3.0	JavaScript 1.1 o superior
Lenguajes de marcado	MXML	XHTML
Estilos CSS	Si	Si
Estilos XML	Si	Si (XSLT)
Lenguajes de Programación	Servidor: PHP, JSP, ASP, ASPX, CFM, Cliente: JavaScript	Servidor: PHP, JSP, ASP, ASPX, CFM, PHYTON, RUBY ON RAILS Cliente: JavaScript
Otras notaciones	No	Si (JSON)
Arquitectura		
Aspectos Gráficos	Si - Flex Charting	Si - SVG
Optimización de las aplicaciones	AS Profile, Runtime Shared Libraries, Flex-	XMLHttpRequest Librerías de código

	Ajax Bridge	abierto
Integridad tecnológica		
De terceros	Tienen gran integridad, ya que es tecnología complementaria a la capa de presentación	
Base de Datos	Puede trabajar con varios motores de base de datos como: SQL, Sybase, Oracle	Puede trabajar con varios DBMS como: MySQL, SQL, Postgress, Informix, Sybase, Oracle
Componentes de integración de aplicaciones	Amplia integración con servidores de aplicaciones como: Apache Tomcat BEA WebLogic WebSphere JBoss Oracle AS	
Sistemas Operativos	Soportado en todas las plataformas	
Navegadores	Soportado en navegadores que incluyan Flash Player	Soportado en navegadores que tengan el objeto XMLHttpRequest
Servicios Web	Si	Si
Costos en infraestructura tecnológica		
Costo o Precio	1 CPU – 10000 usd	Gratis o más de 2000 usd
Soporte de ayuda	Varios recursos como ayuda online, aplicaciones de ejemplo, blogs entre otros	Muchas información como libros digitales, blogs especializados entre otros
Licencias	MVLP	GPL o propietarias

Estándares	HTTP, HTTPS, XML, DOM, CSS, XHTML, Ecma-262, SVG, SOAP.	
Otros		
Amplitud de uso en el desarrollo de software	Reducido por los altos costos	En crecimiento porque a cambiado la forma para desarrollar aplicaciones Web
Facilidad de uso y programación	Gracias a su IDE el desarrollo de código resulta fácil	Se requiere sólidos conocimientos de las tecnologías que usa Ajax razón por la cual es de complejidad avanzada si se desea desarrollar aplicaciones altamente interactivas
Documentación	Si (normal)	Si (amplia la mayoría de nivel muy básico)

Tabla 6.1 Resumen comparativo teórico

A continuación se presenta una tabla resumen de la comparativa de las ventajas y desventajas entre Flex y Ajax:

AJAX	FLEX
VENTAJAS	
Todos los componentes y estándares son de código abierto.	Adopta las herramientas, flujos de trabajo y las infraestructuras

	existentes de terceros.
Complemento en las capas básicas de la arquitectura cliente-servidor.	Flujo de trabajo programático declarativo basado en XML.
Causa-efecto: el flujo de navegación lo va creando el usuario, se evita una capa de control de flujo (J2EE).	Modelo basado en normas XML, SOAP y ECMAScript.
Implementa validaciones en el lado servidor por motivos de seguridad, como las Aplicaciones tradicionales.	Integración con estructuras existentes como EJB, Struts entre otros
Enviar peticiones al servidor Web para traer solamente los datos que son necesarios.	Biblioteca extensa y dinámica de elementos para crear interfaces de usuario
El resultado es una interfaz con mayor respuesta, fluida y en menor tiempo de procesamiento en el servidor Web.	Servicios durante la ejecución de alto rendimiento
Los métodos de carga asincrónicos reducen el cuello de botella. Este método utiliza múltiples conexiones en vez de la única que usan muchas páginas Web.	Depuración y perfilado completo
Se adiciona una capa lógica de presentación para gestionar los eventos que se transforman en llamadas al servidor, actualizando la interfaz.	Permite crear documentos Flash con XML y Actionscript.
Se han desarrollado muchos toolkits, librerías y frameworks.	Es compatible con Windows, Solaris, Linux, Servidores J2EE y Flash Player.
No se requiere ningún plugin o software adicional.	Utiliza a Flash Player como parte de su Arquitectura
DESVENTAJAS	
Mantenimiento de código: El código de AJAX tiene la tendencia a ponerse complicado mientras crezca el proyecto.	La mayoría de empresas no utiliza esta tecnología por ser costosa una implementación completa esta sobre los 7.000 dólares, lo que afecta a las PYMES de forma directa ya que

	pueden optar por otras opciones tecnológicas que tendrán los mismos resultados.
Usabilidad, los botones del navegador "Atrás" y "Refrescar" pueden funcionar mal en algunas ocasiones, ya que aunque el contenido cambia dinámicamente, la página sigue siendo la misma.	El IDE no permite usar lenguajes como C# de .NET.
Dificultad en encontrar errores (debug), el procesamiento lógico de las aplicaciones Ajax se encuentra entre el servidor y el cliente.	Toda la aplicación debe ser compilada en la primera solicitud HTTP, lo cual produce lentitud el primer acceso a la aplicación.
Posibles problemas de seguridad, Ajax por ser algo nuevo puede ser susceptible a los ataques de hackers, si no se sabe usarlo correctamente.	Estar atados a una sola casa fabricante de herramientas ya que la integridad con nuevas tendencias tecnológicas en Web 2.0 están descartadas como por ejemplo integración con JSF, XUL, JSON o Ruby on Rails.
Accesibilidad, existe el riesgo de que algunos usuarios no puedan usar aplicaciones Ajax ya que se basan en Javascript.	Para que funcione correctamente se necesita mayor cantidad de recursos en hardware por ejemplo: Procesador Pentium IV o similares, 1GB de memoria, 400 MB de espacio es disco.

Tabla6.2 Cuadro comparativo de ventajas y desventajas de Flex y Ajax

A continuación se presenta una tabla resumen referente a la seguridad entre Flex y Ajax:

VULNERABILIDAD	SOLUCION
AJAX	
Scripting cruzado o Cross-Site Scripting (XSS)	Filtrar las etiquetas no deseadas de los datos de la entrada

	<p>FLEX ur los caracteres especiales de las etiquetas</p> <p>Protejer las sesiones</p>
Peticiones falsificadas o Cross-site request forgery (CSRF)	<p>Confirmación para procesos importantes usando un token de tiempo</p> <p>Confirmación para procesos importantes usando reautenticación</p>
Rechazo del Servicio (DoS)	<p>Use retrasos para ahogar las peticiones</p> <p>Optimizar las respuestas de datos Ajax</p>
Protección de la propiedad intelectual y la lógica comercial	Ofuscación del JavaScript
Ataque en el cliente	<p>No escribir código ActionScript con sentencias de validación o solicitudes SQL</p> <p>Usar un medio seguro para el transporte (SSL).</p>
Ataque en la transferencia	Verificar la confiabilidad y la calidad de sus servicios Web prestados por terceros
Ingeniería inversa sobre el archivo .swf	<p>Los datos nunca deberían ser parte del código compilado dentro de un archivo .swf.</p> <p>Si es inevitable la inclusión de los algoritmos más privados, ofuscar el código, con herramientas como ActionScript Obfuscator.</p>

Tabla6.3 Cuadro comparativo de vulnerabilidades y soluciones de Flex y Ajax

6.4 ANÁLISIS COMPARATIVO PRÁCTICO

6.4.1 ADOBE FLEX

Para desarrollar aplicaciones enriquecidas con Adobe Flex debemos instalar los componentes de su arquitectura, teniendo los siguientes:

- Flash Player 7.0.14 o superior
- Flex Builder 2
- ColdFusion Extensions para Flex Builder 2
- Flex Data Services Express
- Flex Charting

Es importante analizar los requerimientos que demanda esta tecnología:

Flex Builder 2 (incluido Charting y SDK) para Windows

- Procesador Intel® Pentium® 4
- 1GB de RAM
- 300MB de espacio libre en disco
- Java™ Máquina Virtual: Sun JRE 1.4.2 (incluido), o superiores
- Opcionalmente tenemos plug-ins para Eclipse 3.1 o 3.2, BEA Workshop 3.2.1, o IBM Rational Software Architect 7.0

Navegadores

Microsoft Internet Explorer 5.0, Mozilla Firefox 1.x, Netscape 7.x, AOL 9, Opera 7.11 o superiores.

Se incluye en el Anexo 6.1 la Instalación.

6.4.2 AJAX

Para desarrollar aplicaciones enriquecidas con Ajax tenemos una gama muy amplia de arquitecturas y lenguajes de programación Web como se ha venido hablando. Ya que nos permite elegir el lenguaje del servidor para la lógica comercial, sin embargo

por que elegir AMP (Apache, MySQL, PHP) y no ambientes más robustos como (ASP.NET, JSP) por varias razones:

- Mayor integración con JavaScript explotando la funcionalidad, diseño e ideología Ajax.
- Integridad en entornos de desarrollo Microsoft ya que podemos usar Internet Information Server y trabajar con ASP sin problemas.
- Tanto PHP y JavaScript son tecnologías de código abierto que son utilizadas por muchas comunidades de desarrollo.
- Son lenguajes del modelo orientado a objetos que al fusionarlos obtendremos magníficos resultados.
- Evitar gastos de licencias de sistema operativo, servidor Web, base de datos y herramientas de desarrollo.
- Extender la vida útil de sus servidores antiguos aprovechando los requerimientos de recursos más modestos que requiere esta instalación.
- Evitar gastos de librerías o componentes externos para las aplicaciones Web, PHP y JavaScript puede generar sin costo adicional gráficas, PDF's, ente otros documentos.

Los componentes para instalar esta arquitectura, son los siguientes:

- Apache 1.3.33 (Win32) o superior
- PHP 4 o superior
- MySQL 5.0.16 o superior
- Librerías JavaScript código abierto:
 - Prototype 1.4.0 o superior
 - Rico 1.1.2 o superior
 - Scriptaculous.
- Navegadores Modernos (soporte de XMLHTTP) como: Microsoft Internet Explorer 5.0, Mozilla Firefox 1.x, Netscape 7.x o superiores

Funcionalidades PHP más relevantes al desarrollar con Ajax:

- Programación orientada a objetos

- Parser de documentos XML
- Funciones para manejo de directorios
- Funciones de encriptación de datos
- Manejo de cuentas de mail IMAP y POP3
- Acceso a bases de datos (Mysql, Oracle, Postgress, Sybase, entre otras)
- Manejo de sesiones

Se incluye en el Anexo 6.1 la Instalación.

6.4.3 PROTOTIPO MÍNIMO PARA DEMOSTRACIÓN DE FUNCIONALIDADES AJAX Y FLEX

Para confirmar este análisis se procede a realizar un prototipo mínimo tanto con Flex como con Ajax para determinar su eficiencia, portabilidad, integridad y presentación. Este pequeño aplicativo será la autenticación de un usuario administrador que accederá al ingreso de datos en un formulario y que serán almacenados en una Base de Datos. Para mayor información en el Anexo 6.2 se lo detalla.

A continuación se presenta la tabla resumen de la comparativa práctica entre Adobe Flex y Ajax:

	FLEX	AJAX
Tamaño mínimo de una Aplicación “hola mundo”	107 KBytes	24 KBytes
Alta interacción en la interfaz de usuario	Si	Si
Mínimo de procesador	Intel Pentium 4 / 800MHz	Intel Pentium III / 600 MHz
Mínimo de Memoria RAM	1 GB recomendado	256 MB recomendado
Espacio en disco libre	+400 MB	-400MB
Sistemas Operativos de prueba	Windows Linux	Windows Linux

Tabla 6.4 Resumen comparativo práctico

6.5 CONCLUSIONES PRÁCTICAS MEDIANTE EL PROTOTIPO

Adobe flex:

- Flex permite desarrollar robustas aplicaciones swf pero el primer acceso y descarga a la aplicación es muy pesada por los componentes que son ejecutados.
- Flex tiene excelente integridad con tecnologías de código abierto como PHP
- Flex enriquece la interacción del cliente proporcionándole mecanismos de efectos a sus componentes.
- Flex permite personalizar y desarrollar componentes propios en las aplicaciones.
- En el desarrollo de código Flex permite ahorrar mucho tiempo ya que su IDE tiene herramientas de diseño que permiten observar los componentes de la aplicación.
- Flex nos permite utilizar paquetes .as utilizados en Adobe Flash pero no tiene un depurador y hay que hacerlo a la mano con trace() lo cual es muy tedioso.

Ajax:

- Ajax necesita mayores controles para ofrecer seguridad en las aplicaciones Web ocasionando mayor trabajo a nivel de la aplicación tanto en el lado del cliente como en el servidor.
- Ajax exige a explotar la capa de presentación en una aplicación Web del cliente trayendo beneficios en la interacción de la interfaz, para esto debemos saber como se utilizar los gestores de eventos.
- Ajax exige tener sólidos conocimientos de JavaScript y DOM Nivel 2 para poder desarrollar aplicaciones asincrónicas y proveer de una alta interacción al usuario en la aplicación.
- En las aplicaciones Ajax el usuario muchas veces desconoce lo que esta sucediendo en la aplicación, por este motivo debemos encontrar formas para hacer feedback e informarle al usuario lo que esta pasando.

- La arquitectura de las aplicaciones Ajax es variante en la capa de la lógica del negocio y por ende se adapta y tiene alta integridad a cualquier entorno de desarrollo.
- Ajax nos permite trabajar con renderización de objetos en un documento Web y no con páginas enteras.
- Optimización alta de las funcionalidades que permiten utilizar los navegadores Web como CSS, JAVASCRIPT (XMLHTTPREQUEST, BOM, DOM, JSON) XML, XSLT, WEB SERVICES entre las principales al usar Ajax.

6.6 RESULTADOS DEL ANÁLISIS COMPARATIVO

Una vez que se han analizado tanto, la parte teórica como práctica, es tiempo de definir una representación de los indicadores expuestos anteriormente para asegurar que tecnología es la más adecuada para desarrollar una RIA.

Indicadores	Valor
Malo	0
Regular	1
Bueno	2
Excelente	3

Tabla 6.5 Valoración análisis comparativo

FACTORES		VALOR DE LA EVALUACIÓN	
		FLEX	AJAX
A	Entornos de desarrollo	3	2
B	Integración con lenguajes de programación Web	3	3
C	Integridad tecnológica con terceros	2	3
D	Costos en infraestructura tecnológica	0	2
E	Soporte de Estándares 3WC	3	3

F	Seguridad	2	1
G	Amplitud de uso en el desarrollo de software	1	3
H	Facilidad de uso y programación	2	1
I	Alta interacción en la interfaz de usuario	3	3
J	Mantenimiento coste y soporte hardware	1	3
K	Orientación a Objetos	3	3
L	Acceso a Base de Datos	2	2
M	Sistemas Operativos	3	3
TOTAL		28	32
PORCENTAJE		71.79	82.05

Tabla 6.6 Resultados análisis comparativo

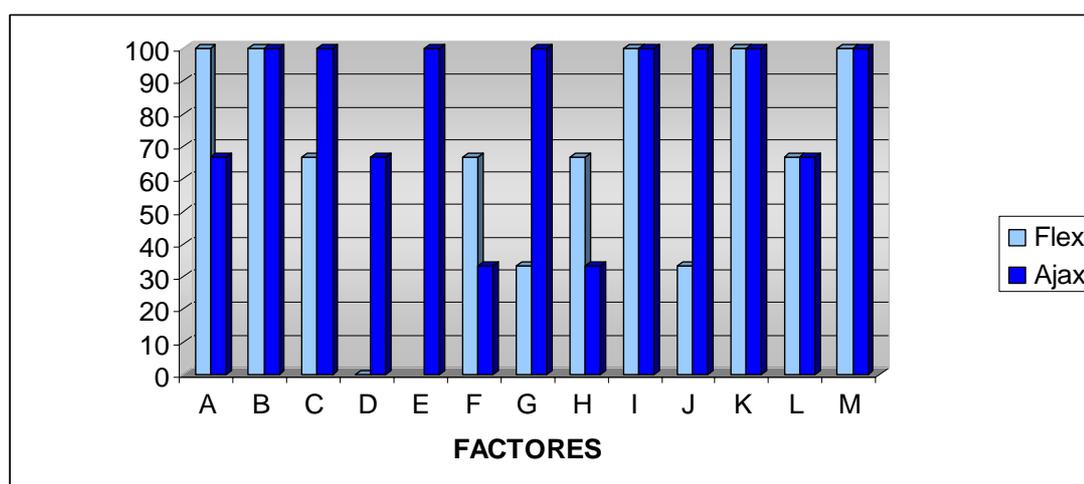


Figura 6.3 Resultados análisis comparativo

Como hemos observado y analizado las dos tecnologías cumplen con excelentes requisitos para desarrollar RIA's, por tal motivo depende de las características y necesidades de un proyecto Web para usar cualquiera de las dos.

En nuestro caso en particular para el Colegio de Arquitectos del Ecuador sede Imbabura, conocidas todas las características de este análisis, la tecnología que mejor se ajusta a las necesidades para el desarrollo de la aplicación Web es **AJAX** por su flexibilidad, portabilidad y en especial bajo costo económico.