Universidad Católica Nuestra Señora de la Asunción

Facultad de Ciencias y Tecnología

Ingeniería Informática

Teoría y Aplicación de la Informática 2

Profesor: Juan Urraza

Aplicaciones de Internet enriquecidas para escritorio

César Paredes

Setiembre de 2009

Indice

Rich Internet applications for desktop

(Aplicaciones de internet enriquecidas para escritorio)

1. Introducción

Dos clásicos problemas de la informática han sido históricamente como aprovechar mejor los recursos de hardware, buscando el mejor desempeño de las aplicaciones; y en qué manera presentar la información al usuario, buscando que el software sea más amigable.

Los navegadores Web nacieron con la finalidad de facilitar el acceso a la información disponible a través de las grandes redes, principalmente Internet. Los navegadores, aprovechándose de las mejoras en la interfaz grafica de los sistemas operativos que surgían a principios de los 90, basaron su funcionalidad en texto e imágenes presentadas en una "ventana", utilizando principalmente el lenguaje HTML.

El rápido crecimiento de la Internet, y las constantes mejoras de los navegadores Web vienen de la mano, a medida que los navegadores se volvieron más amigables, y disponibles en todos lados, más gente se sintió atraída a utilizarlos, inclusive usuarios que nunca se habían atrevido a utilizar una computadora.

Basándose en la ubicuidad, portabilidad y amigabilidad de los navegadores web, a partir de finales de los 90 se va fortaleciendo el concepto de aplicaciones Web, estas son aplicaciones, similares a sus "versiones de escritorio", pero normalmente con funcionalidades limitadas, como email basado en web, traductores de texto, etc.

Varias clásicas aplicaciones de escritorio también se vieron influenciadas por esta corriente, es así como las típicas aplicaciones empresariales basadas en bases de datos pasaron a utilizar el navegador web como forma de presentación e interacción con el usuario.

Ambos fenómenos fueron impulsados por una cantidad de lenguajes de programación orientados a la web, y el hecho que al mismo tiempo los navegadores web fueron ganando "aptitudes", principalmente en lo que se refiere al tipo de contenido que eran capaces de mostrar, al texto e imágenes se les agrego sonido, animaciones basadas en imágenes vectoriales, videos. Todo esto dentro del concepto de la Web 2.0, donde también surge el término Rich Internet Application (RIA), o aplicación de internet enriquecida, que hace alusión a una evolución de la tradicional web application.

Actualmente se pueden encontrar todo tipo de aplicaciones en la Web y RIAs, casi cualquier aplicación de escritorio tiene su "reemplazante Web", muchas veces gratis. Se pueden mencionar editores de texto comparables en su funcionalidad con los ofrecidos por Microsoft, editores de imágenes similares a los ofrecidos por Adobe, juegos en línea, etc.

Sin embargo las aplicaciones contenidas dentro del navegador Web se enfrentan a grandes limitaciones. Entre ellas se puede mencionar principalmente dos, la dependencia de la conexión a Internet, que todavía no es tan ubicua como sería deseable. Y la falta de mejor interacción con el hardware, que limita tanto las funcionalidades que estas pueden ofrecer, como el rendimiento de las mismas, y la amigabilidad de la interfaz humano-computadora.

En la búsqueda a soluciones a dichos problemas surge la propuesta, que hasta parecería un poco contradictoria, de hacer que las aplicaciones de internet enriquecidas sean ejecutables como aplicaciones de escritorio. Esto implicaría una interacción más directa con el sistema operativo, y por ende mejor aprovechamiento de los recursos de hardware; y también una cierta independencia de la conectividad a la Internet.

Dichas propuestas ya han tomado forma en plataformas tecnológicas ya disponibles para el desarrollo y despliegue de las llamadas "aplicaciones web para escritorio", que van ganando cada vez más aceptación, y se perfilan como algo que formara parte de nuestro repertorio típico de aplicaciones en los siguientes años.

En este trabajo se presentaran dichas plataformas tecnológicas, haciendo una comparativa entre ellas con el fin de ampliar nuestros conocimientos y poder hacer una decisión informada a la hora de decidir cual plataforma utilizar para el desarrollo de nuestra próxima aplicación Web.

2. Una primera propuesta: extender el navegador

El modelo de arquitectura típicamente seguido por las aplicaciones Web se basa en un Servidor que ofrece la aplicación a varios clientes. La aplicación está contenida dentro del navegador Web, y la interacción entre el servidor y el cliente se realiza principalmente por medio de eventos HTTP.

En este modelo la aplicación web está limitada a utilizar tanto para la presentación como para las funcionalidades, los elementos que el navegador web le permita. En el mejor de los casos, por medio de tecnologías como Flash y Java script, se permite también que la aplicación ejecute parte del código localmente, sin necesidad de conectarse al servidor.

Fig 1: Diagrama de la clásica arquitectura Cliente Servidor

Es fácil notar que un usuario que utilice mayoritariamente aplicaciones Web convierte a su navegador Web como en un segundo sistema operativo. Es decir, se podría establecer un modelo de capas que represente la situación, donde la capa mas baja es el sistema operativo, que solo interactúa con la siguiente capa, que es el navegador Web, que a su vez contiene a las aplicaciones utilizadas por el usuario.

Ese fenómeno lleva a una situación donde la gran cantidad de ventanas o pestañas del navegador web vuelven a la interfaz humano computadora bastante engorrosa, además de dejar sin aprovechar las bondades de los sistema operativos actuales que facilitan el multi tasking en todos los niveles.

Para paliar dicha situación, surgió una primera propuesta, impulsada principalmente por empresas dedicadas a Internet, como Mozilla y Google. Dicha propuesta se basa en hacer que aplicaciones Web puedan ser ejecutadas en forma más similar a las aplicaciones de escritorio, pero aun residiendo en el navegador.

La propuesta ataca principalmente el problema de la presentación de dichas aplicaciones, la facilidad de su uso y su contexto, evita que aparezca el navegador y sus menús, plug-ins y pestañas que nada tienen que ver con la aplicación en sí, dejando una ventana exclusivamente para la aplicación Web, ahora más parecida a una de escritorio. Inclusive es posible crear accesos directos a las aplicaciones, agregando así más transparencia y amigabilidad.

En cuanto a lo que se refiere a la independencia de la aplicación a la conectividad de internet, no se toman medidas, más bien se deja dicho aspecto a cargo de quien desarrolle el software, sugiriendo siempre utilizar tecnologías que permitan almacenar la aplicación y

los datos necesarios para su ejecución localmente, y conectarse a internet solo cuando sea imprescindible la transferencia de datos.

Con respecto al rendimiento, los que plantean esta propuesta argumentan que existen mejoras en el rendimiento gracias a que la aplicación se ejecuta dentro de un proceso del navegador exclusivo para dicha aplicación, con solo lo específicamente necesario, no con todos los agregados ni funciones innecesarias.

Entran dentro de un concepto bastante nuevo llamado Site Specific Browser, que como su nombre lo dice, son navegadores que ejecutan específicamente un solo sitio. Dentro de esta categoría se destacan el Mozilla Prism, Google Chrome, Bubbles y otros.

2.1 Mozilla Prism

Anteriormente conocido como WebRunner, actualmente se encuentra en fase beta, disponible para descarga y libre utilización. Está basado sobre el mismo motor del Mozilla Firefox, y corre como una extensión del mismo, sobre sistemas operativos Windows, Linux y Mac.

Permite separar cualquier aplicación Web del navegador a nivel del proceso propiamente dicho, esto tiene algunas ventajas intrínsecas que lo hacen interesante porque, por ejemplo, si se cierra el browser, la aplicación está corriendo separada sin problemas.

Algunas de sus funcionalidades destacadas por Mozilla:

- Crear accesos directos a aplicaciones de internet
- Ejecutar aplicaciones automáticamente con el inicio del sistema operativo
- Minimizar al símbolo de sistema (tray).
- Mayor estabilidad al deshabilitar plug ins inestables
- Alertas emergentes

Fig 2: www.twitter.com ejecutándose utilizando Prism

Fig 3: Barra de tarea de Windows, con aplicaciones web ejecutándose automáticamente al inicio del sistema operativo, utilizando Prism

2.2 Google Chrome

Chrome es un navegador Web desarrollado por Google, con el motor WebKit, está en fase beta, es de código abierto, compatible con sistemas operativos Windows, Linux y Mac, cuenta con características innovadoras en cuanto a diseño, tales como la ejecución de cada sitio como una nueva instancia del navegador a nivel de procesos del sistema operativo.

Entre sus funcionalidades se encuentran las “Ventanas de aplicaciones”, que permiten ejecutar sitios web como aplicaciones independientes del navegador, en el escritorio. Crear una nueva ventana de aplicación es tan simple como crear un acceso directo a la página utilizando el URL de la misma.

Fig 4: Un acceso directo a www.gmail.com que se está ejecutando como aplicación de escritorio en Google Chrome

3. Otra propuesta: entornos de desarrollo y ejecución

Mientras la primera opción se presenta básicamente como una extensión a los navegadores web, otorgándoles la habilidad de “convertir” aplicaciones web a aplicaciones de escritorio, otra solución, planteada por gigantes del software como Sun, Microsoft y Adobe, además de lo ofrecido por los anteriores, implica cambios más radicales, desde el desarrollo hasta la ejecución de la aplicación de internet enriquecida.

Adobe Air, JavaFX, y Microsoft Silverlight tienen diferencias bastantes marcadas, pero en el fondo tambien tienen notables similitudes, como el hecho de que son entornos de desarrollo y ejecución para aplicaciones de internet enriquecidas, que buscan mejorar las prestaciones de las aplicaciones basadas en navegadores Web, tanto en su aspecto funcional como de presentación.

En cierta forma, sustituyen al navegador Web como entorno de ejecución, dando mayores libertades a la aplicación Web, como acceso a aceleración por hardware, almacenamiento local, soporte para mayor diversidad de contenido, etc.

Este tipo de aplicaciones normalmente requieren una pequeña instalación inicial en el disco duro local, y pueden ser ejecutadas mayormente sin conexión a internet, ya que cuentan con los datos necesarios para el uso típico localmente, gracias a sistemas de bases de datos integrados en el ambiente de ejecución.

Estas plataformas están ganando rápida aceptación tanto en el ambiente de los desarolladores de software como de usuarios, principalmente gracias a su fácil distribución y notable portabilidad.

Si bien fueron pensadas con la idea de ser un “hibrido” entre aplicaciones de Internet enriquecidas y aplicaciones de escritorio, es el segundo sector el que está creciendo más rápidamente, a veces ignorando por completo al navegador Web.

En la actualidad están surgiendo aun más plataformas con un enfoque similar al mencionado, pero considerando el éxito ya obtenido, y los grandes nombres que las respaldan analizaremos más profundamente tres opciones disponibles, Adobe Integrated Runtime, Sun JavaFX y Microsoft Silverlight.

3.1 Adobe AIR

El Adobe Integrated Runtime (AIR) es un ambiente de ejecución que permite utilizar populares tecnologías Web como Flash, Actionscript y AJAX para desarrollar aplicaciones que se ejecutan en el escritorio, con portabilidad a varias plataformas como Linux, Windows, Mac.

Adobe AIR incluye un motor de base de datos SQLite para almacenamiento de datos en el equipo del usuario, con soporte para encriptación de datos y firmas digitales para conexiones seguras a través de Internet, todo esto, siempre independientemente del navegador Web.

Para el desarrollo de aplicaciones AIR Adobe ofrece un SDK que tiene una notable flexibilidad, es capaz de interactuar con varios editores HTML. Texto plano, o Ajax, y con la mayoría de productos Adobe, como el Dreamweaver, Flash, Flex Builder, AfterFX, etc.

Fig 5: uvLayer, una aplicación desarrollada con Adobe AIR, con el fin de almacenar y compartir fotografías y videos.

3.2 Sun JavaFX

JavaFX es una tecnología desarrollada por Sun Microsystems para permitir la ejecución de RIAs en una gran variedad de dispositivos, ya sea dentro de un navegador Web o independientemente de este.

JavaFX está basado en tecnologias ya existentes como la Java Virtual Machine, que le permite portabilidad a casi cualquier dispositivo que soporte el Java Runtime Environment, como sistemas Windows, Mac, y varios sistemas operativos para dispositivos móviles. Actualmente el soporte para sistemas Linux esta en desarrollo.

Para el desarrollo de aplicaciones JavaFX, Sun ofrece un SDK con varias herramientas para gráficos 3d, sonido, animaciones, etc. El lenguaje de programación es el JavaFX Script, que tiene una sintaxis muy similar al JavaSE. También existen cartuchos que permiten importar imágenes y animaciones desarrolladas con otras herramientas como Adobe Illustrator o Media Factory.

Una característica particular de JavaFX es que permite al usuario una instalación muy sencilla. "Drag-to-install" permite simplemente arrastrar una aplicación desde el navegador hasta el escritorio, volviendo la aplicación Web totalmente independiente del navegador sin que esta pierda su estado o contexto actual.

Fig 6: Las capas que constituyen la arquitectura aplicaciones de javaFX

3.3 Microsoft Silverlight

Microsoft Silverlight nació como un plug in para navegadores Web que permita desarrollar aplicaciones enriquecida de Internet, que fuera portable también a dispositivos móviles basados en el sistema operativo Windows Mobile, pero fue evolucionando para convertirse en una herramienta con muchas más capacidades. Siendo portable a la mayoría de los sistemas operativos Windows y al Mac OS X.

Silverlight tiene soporte para gráficos vectoriales, animaciones, audio y video. Su desarrollo es logrado mediante el Silverlight Toolkit y la plataforma .NET, utilizando cualquier IDE que soporte dichas tecnologías, como el Microsoft Visual Studio 2008 o Eclipse. Entre sus funcionalidades resalta la ventaja de permitir utilizar aceleración por hardware utilizando funciones de directX.

Actualmente la versión 3.0 trae como una de sus innovaciones la posibilidad de que aplicaciones Web desarrolladas con Silverlight puedan ser ejecutadas fuera del entorno del

navegador, como una aplicación de escritorio, en modo sin conexión a internet, o también como un “gadget” de Windows Vista.

Fig 7: Demo de aceleración 3d utilizando Silverlight en un navegador Web

4. Para que utilizarlos... donde se ubican?.

Considerando que todas estas son todavia tecnologias emergentes, que no estan completamente establecidas como standares del mercado, uno se podria preguntar si vale la pena utilizarlas, o si son mas de lo mismo y seria mejor quedarnos con nuestra plataforma preferida para desarrollo web.

Es dificil decir si todas ellas pueden coexistir, o si una terminara prevaleciendo, pero es necesario aclarar que no son similares a tecnologias actualmente estandarizadas como .NET o JavaEE o alternativas similares, son mas bien complementarias, y hasta en algunos casos extensiones de las mismas. Tienen como objetivo permitir hacer cosas que anteriormente no se podian hacer o era muy complejo y consumidor de tiempo realizarlo.

El lugar exacto donde pretenden ubicarse estas plataformas es conocido en el ambito IT como herramientas que permiten el desarrollo de Aplicaciones para Internet Enriquecidas (RIAs). A esto cada una de las opciones le añade sus particularidades y lo que consideran que pueda llegar a ser mas util tanto para el desarrollo, entrega y utilizacion de la aplicacion.

Desde el punto de vista del usuario final, las diferencias principales que podra notar en aplicaciones que utilicen estas tecnologias seran la independencia del navegador, dando mayor amigabilidad a las aplicaciones y la presentacion mas enriquecida a nivel estetico, y posiblemente mejoras en el rendimiento por el mejor aprovechamiento de recursos del sistema.

4.1 Productos similares, enfoques diferentes

Dando un rapido vistazo a las caracteristicas y capacidades de Adobe AIR, Microsoft Silverlight y Sun JavaFX podemos encontrar varias similitudes, esto ocurre naturalmente porque son todos productos que compiten en una misma categoria, (la de herramientas para desarrollo de RIAs).

Analizando con mayor detenimiento y considerando la trayectoria de las empresas detras de dichas tecnologias podemos darnos cuenta que realmente no son tan similares como parecen, o por lo menos, no estan orientados al mismo tipo de usuario (en este caso desarrolladores de aplicaciones).

Adobe es conocida por sus productos orientados al manejo de imagenes y animaciones web como Photoshop, Illustrator, Dreamweaver y Flash, utilizados mayoritariamente por diseñadores graficos y diseñadores web. Mientras tanto, Microsoft y Sun (Java) son mas conocidos por sus plataformas para aplicaciones de escritorio y sistemas web (.NET, JavaEE, JavaScript) utilizadas por desarrolladores de software.

4.2 Diseñadores, ya no necesitan del navegador.

No nos debe sorprender que Adobe dirija AIR a diseñadores web, que estan acostumbrados a trabajar con las tradicionales herramientas como Dreamweaver y Flash, y ya conocen el lenguaje de programacion ActionScript utilizado para AIR. En este caso, lo que se le ofrece al desarrollador es la posibilidad de dejar de depender del navegador Web, y poder explorar nuevas fronteras utilizando mas recursos de software y hardware, como bases de datos, graficos 3d, almacenamiento local, etc.

Podemos ver claramente lo que nos dice Adobe, en su documentacion, seccion getting started:

> " *This is easier than using lower level languages such as C and C++, and does away with the need to learn complex low-level APIs specific to each operating system.* "

4.3 Expandiendo el territorio.

En el caso de Silverlight, Microsoft plantea inicialmente entrar al territorio de animaciones de imagenes vectoriales, compitiendo directamente con Flash, pero luego decide integrar Silverlight con .NET y asi extender el "Windows Presentation Foundation" que es basicamente la alternativa que Microsoft nos ofrece para la capa de presentacion de aplicaciones .NET.

Considerando que el principal lenguaje de programacion para la logica del programa en .NET sigue siendo C#, con algunas alternativas permitidas como IronRuby o IronPython, y la necesidad de conocer todo el framework en general, esta alternativa puede resultar mas atractiva para desarrolladores de sistemas web y aplicaciones de escritorio, que pueden aprovechar Silverlight para enriquecer la presentacion y dar mas amigabilidad a sus aplicaciones.

4.4 Flexibilidad por sobre todo.

Java ha sido desde sus inicios un lenguaje que busca la flexibilidad en todo sentido, como en su lema "write once, run everywhere" busca ser compatible con todas las plataformas posibles, y al mismo tiempo tiene sus distintas versiones (SE, ME, EE, JavaScript) que son especializaciones del lenguaje orientadas a cierto tipo de aplicaciones.

Sun, en su presentacion, resalta la portabilidad de JavaFX como una de sus principales cualidades que lo hacen mejor que sus competidores, ademas de la flexibilidad que provee a traves de su virtual machine que permite que el mismo codigo que se ejecuta dentro del navegador pueda ser ejecutado como aplicacion de escritorio con un simple "drag-to-install". Los fanaticos de Java estaran felices de tener su propia plataforma para desarrollo de RIAs.

4.5 Aplicaciones de Internet, trabajando sin conexión?

A pesar de que parezca algo paradojico, todas estas tecnologias presentan como una de sus innovaciones la posibilidad de trabajar sin conexion a Internet. Obviamente esto presenta grandes limitaciones para aplicaciones que requieren continuo intercambio de datos con un servidor online, que puede ser parcialmente solucionado por sistemas de almacenamiento local, que suben o bajan los datos cuando exista conexion a la red.

No hay que olvidar, que tanto AIR, Silverlight y JavaFX no se centran solo en aplicaciones online, sino mas bien expanden las posibilidades a aplicaciones que normalmente se encuentran en el escritorio, como reproductores multimedia, juegos 3D, organizadores, etc. Es ahi donde la caracteristica de poder trabajar sin conexion a internet se vuelve mas interesante, y cuando exista conectividad, se presentan con mayores funcionalidades utilizando web services o actualizaciones.

5. Conclusión

Está claro que las tecnologías disponibles actualmente para la ejecución de aplicaciones de internet enriquecidas están aun en desarrollo, y tienen mucho por ofrecer.

Si bien la opción planteada por los navegadores y sus extensiones parece bastante básica, no deja de ser interesante porque permite a aplicaciones Webs ya existentes ejecutarse como aplicaciones de escritorio, sin ninguna modificación.

Las opciones más robustas, que plantean diseñar las aplicaciones para internet como aplicaciones hibridas desde un comienzo ofrecen claramente mayores ventajas, como aceleración por hardware y almacenamiento local.

Este segundo tipo de aplicaciones son en mi opinión las verdaderas aplicaciones de internet enriquecidas, que van borrando lentamente la delgada línea que existe entre aplicaciones de escritorio y Web.

6. Bibliografia

Mozilla Prism, http://prism.mozilla.com/

Google Chrome, http://www.google.com/chrome/intl/es/features.html?hl=es

Adobe Air, http://www.adobe.com/products/air/

Sun Microsystems JavaFX, http://javafx.com/

Microsoft Silverlight, http://www.microsoft.com/silverlight

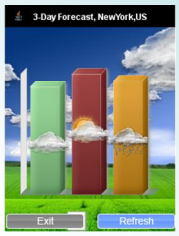
Apendice A

Aplicaciones existentes actualmente (noviembre 2009)

Autor: Jesus Bellasai

Weather Forecasts

- Con la tecnología JavaFX es fácil de construir un software que muestre las previsiones del clima para una zona que el usuario personaliza utilizando Yahoo webservices. Esta aplicación es tanto para la versión desktop como para la versión de dispositivos móviles.



Google Desktop

- Es una aplicación de búsqueda en el escritorio que permite encontrar texto en mensajes de correo electrónico, archivos del equipo, chats y páginas web que se han visitado. Al habilitar su equipo para la búsqueda, el programa le permite acceder fácilmente a su información y le evita tener que organizar manualmente sus archivos, mensajes y marcadores.

Gadgets

- Es un dispositivo que tiene un propósito y una función específica, generalmente de pequeñas proporciones, práctico y a la vez novedoso. Los gadgets suelen tener un diseño más ingenioso que el de la tecnología corriente.
- Algunos ejemplos: Google Gadgets, Microsoft Gadgets, Dashboard software Apple Widgets

LimeWire

- LimeWire es un cliente P2P gratuito para Red Gnutella, diseñado para el intercambio de archivos. Este software funciona en un protocolo abierto, gratuito para el uso público. Fue publicado bajo la licencia GPL de código abierto.
- Está programado en Java por lo que es necesario tener instalado la JRE (Java Runtime Environment).

Vuze

- Antes Azureus, es un cliente de BitTorrent de código abierto y desarrollado en lenguaje de programación Java, por lo que es multiplataforma (teniendo instalada la Máquina virtual Java) y que funciona tanto en sistemas Mac, como Windows o GNU/Linux.