

INTERFACES GRÁFICAS DE USUARIO

Alternativas en GUIs

En este artículo analizamos todas las tecnologías relacionadas con la capa de presentación de una aplicación: XUL, Rich Internet Applications, XAML, AJAX, las soluciones propuestas por Java y la rivalidad Microsoft – Adobe Macromedia.



Cristian Hernán Gaitano Ornia
Especialista en aplicaciones web.
cerge@cerge.net

Como podemos imaginar, el tema que vamos a tratar no sólo interesa a un diseñador web, ya que es curioso ver cómo la mayoría de estas tecnologías se refiere, en mayor o menor grado, al uso directo o indirecto de un término que, cada vez, está más difundido: XML. ¿Estamos ante una nueva era en la programación web?

Es verdad que, a medida que el tiempo avanza, algunas tecnologías reemplazan a otras, así como ciertos conceptos son revisitados o bien mejorados en función de las necesidades que el mercado impone en determinado momento.

En el ámbito tecnológico, vemos que muchas de ellas crecen día a día, mientras que otras tantas son abandonadas o pierden soporte. El sector de las interfaces gráficas de usuario (o, simplemente, GUI) no es la excepción. Dentro del primer grupo es llamativo observar cómo la mayoría de ellas esgrime dos factores generalmente comunes a todas. Uno es el meta lenguaje XML y todos sus dialectos derivados, el cual habitualmente está bien secundado por CSS y JavaScript. El otro es la

notoria necesidad de crear más y mejores experiencias con el sistema cliente y, en la medida de lo posible, reducir al mínimo el desperdicio del proceso de ida y vuelta de información, tan característico del modelo web clásico.

En cuanto al futuro, es muy difícil dar pronósticos e, incluso, puede parecer demasiado subjetivo hacerlo. Las experiencias sugieren que muchas tecnologías han pasado sin pena ni gloria, en tanto que otras, quizá más valorables, fueron presa de diversos factores reinantes. En otras palabras: o bien fueron quedando relegadas al olvido (muchas veces, por haber sido desmesuradamente promocionadas) por el solo hecho de estar dentro de una corriente tecnológica, o bien fueron menos apoyadas en el aspecto comercial.

Mientras tanto, nada mejor que ir avizorando aquellas tecnologías que nacen o que resurgen con el ánimo de establecerse como opciones válidas de desarrollo.

XUL

Para definir a XUL (*XML-based User-interface Language*) con cierta precisión, conviene remontarse al año 1997 y citar como antecedente al motor Gecko, referido generalmente como el “corazón” de los navegadores basados en Mozilla. Este motor de renderización abierto es muy bien reconocido en el ámbito tecnológico por sus diversas facetas de carácter absolutamente portable, extensible, personalizable y multiplataforma. Al no poseer una interfaz de usuario propia (es decir, su función es interpretar el contenido web mostrándolo luego de una forma conveniente), Gecko está siendo usado, actualmente, no sólo por el navegador, sino también para otro tipo de aplicaciones, ya que es totalmente empotrable en una variada gama de productos. Y es aquí donde aparece XUL. En efecto, Gecko se vale de XUL para implementar una determinada interfaz de usuario; veamos cómo lo logra.

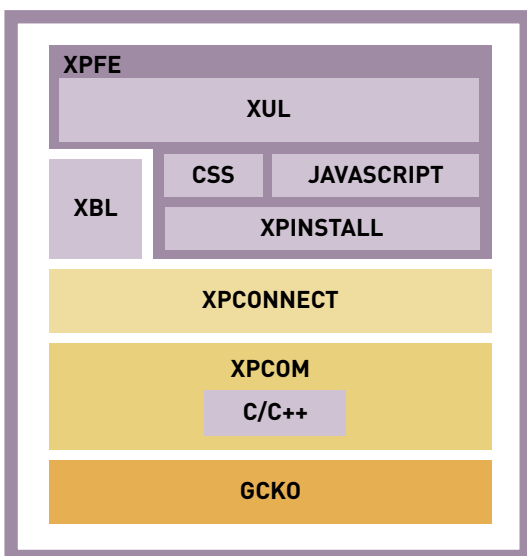
Ya desde su definición misma, XUL se presenta como un lenguaje extensible para la interfaz de usuario de Mozilla. Vale decir, con esta herramienta y el agregado de tecnologías relacionadas al ambiente web –como CSS y JavaScript (las cuales forman XPFE o *Mozilla CrossPlatform Front End* y reemplazan a DHTML)– un desarrollador puede modificar cualquier aspecto en la interfaz de una aplicación Mozilla basada en XUL, poniendo el foco en determinados archivos relacionados de una forma muy similar a como se podrían editar archivos de carácter puramente web.

Este esquema se arma del siguiente modo: mientras XUL sostiene la estructura de la aplicación, JavaScript crea la funcionalidad para una solución Mozilla, CSS se encarga de configurar el diseño y la presentación, y XPInstall (*Cross-Platform Install*) se ocupa de gestionar las instalaciones en diversas plataformas.

Por otra parte, con XUL es posible determinar que una aplicación sea ejecutada desde un servidor web, o bien hacerla totalmente independiente del navegador utilizando XPCOM, la arquitectura de componentes de Mozilla (ver [Figura 1]). Si bien XUL es una tecnología que tiene varios años de trabajo y desarrollo, aún no es un estándar para todos los browsers, y gran parte de su proyección futura está supeditada al crecimiento del grupo de navegadores basados en Mozilla, así como también a que el mercado de los navegadores se expanda y pueda crecer en forma más balanceada y menos monopólica.

Java y un mundo de incontables herramientas

Dar con un solo producto reinante para una determinada tarea en el ambiente Java es casi imposible. Y si la búsqueda empieza en el área de tecnologías



[Figura 1] La arquitectura Mozilla crece en forma gradual, desde el motor Gecko hasta XPFE.



relacionadas a XML –espacio donde Java tradicionalmente se mueve como pez en el agua–, esa tarea se vuelve ardua y singular. Siempre hay varias tecnologías para elegir, y por eso vemos surgir y consolidarse a diario una buena cantidad de APIs destinadas, específicamente, a la generación de GUIs basadas en XML o, incluso, directamente apoyadas en el concepto de XUL. Entre las más representativas, podemos citar a SwingML, Swixml, Jelly, Luthor XUL, Xulux, Thinlets y XWT.

Por mencionar sólo un caso, SwixML trabaja en forma muy efectiva generando interfaces gráficas de usuario a partir de definiciones en archivos XML. Sin embargo, hay que aclarar que esta API se centra, más que nada, en la etapa de presentación, y no es tan permeable a la introducción de interacciones o eventos.

Inmerso en el mundo Java, no podemos olvidarnos de un decisivo aliado como Oracle, que tercia en esta disputa interna de tecnologías aportando su tecnología ADF UIX, orientada, específicamente, a dar soporte en la creación de interfaces web o Java Client, y teniendo siempre en foco el ya mencionado estándar de Java Server Faces (JSF).

Con UIX, Oracle promete desarrollar aplicaciones web con un modelo de componentes XML en la capa de presentación, incluyendo tanto librerías de clases Java y APIs, como analizadores (o parsers), que permitan crear rápidamente productos web orientados a e-learning, catálogos y plataformas de negocios.

RIAs (Rich Internet Applications)

Existen algunas acepciones distintas del término RIA (*Rich Internet Application*), forjado por Macromedia poco tiempo atrás. Por eso, es conveniente aclarar el sentido real de esta nueva generación de soluciones que promete remover la estructura web actual.

Esencialmente, lo que se busca con una RIA no es sólo prescindir en forma definitiva de las cadenas que impone el HTML. Si bien la finalidad central parecería ser proporcionar una mejor y más grata experiencia visual al usuario web desde un punto de vista más técnico, es también su objetivo unificar diferentes plataformas de ejecución desde la capa de presentación.

Macromedia Flex es un producto emergente de avanzada que se encuadra dentro de lo que, actualmente, se denominan RIAs, y que confirma lo que se venía hablando hace mucho tiempo atrás. Es decir, con el paso del tiempo, las aplicaciones basadas en Flash crecieron en cantidad, complejidad y calidad, a lo que sumaron manejo avanzado de datos, controles más complejos y mayor procesamiento en servidor. Con esto fueron arrastrando a muchos diseñadores gráficos a convertirse en auténticos desarrolladores web integrales.

La esencia de Flex es utilizar esa potencialidad en el mercado profesional ofreciendo una combinación de diversas herramientas, con el fin de crear y de reutilizar complejos componentes en formato Flash.

Por otro lado, vemos que Open Lazlo se esgrime como “rival” directo del producto estrella de Macromedia. De hecho, Flex y Open Lazlo toman el mismo camino, pero con distintos vehículos. La otrora *Lazlo Presentation Server* (LPS) –ahora Open Source y rebautizada Open Lazlo– es una atractiva arquitectura que forma parte del mismo segmento de mercado, y que representa uno de los proyectos más interesantes del último tiempo, que conviene seguir muy de cerca.

Ahí donde Flex usa MXML (*Macromedia Flex Markup Language*), OL se vale de LZX como lenguaje de presentación, un muy rico dialecto que soporta POO y permite desarrollar aplicaciones basadas en XML (del cual deriva, en gran medida, LZX), JavaScript (donde Flex usa ActionScript) y la especificación XPath.

En el ámbito de los Web Services, mientras que ambos productos coinciden en SOAP, Flex se centra en un protocolo propio llamado AMF (*Action Message Format*), y Open Lazlo incluye y da muy buena cabida a XML-RPC.

A medida que el tiempo avanza,
algunas tecnologías reemplazan
a otras, así como ciertos conceptos son
revisitados o bien mejorados en función
de las necesidades del mercado.

Adobe y Microsoft en terrenos inversos

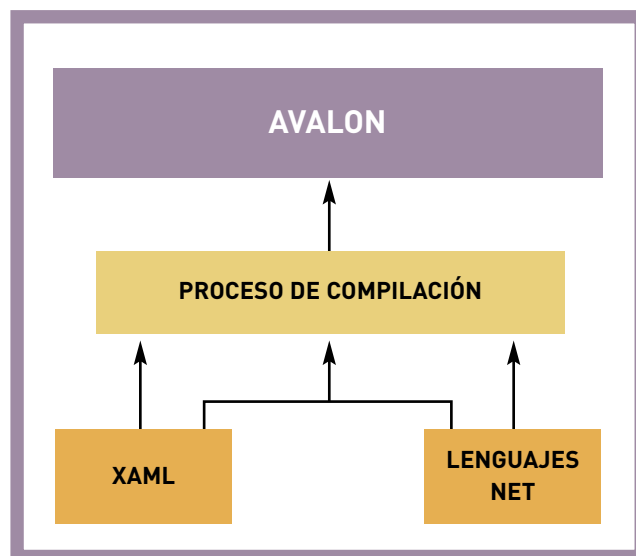
Como imaginarán, Adobe-Macromedia apunta mucho más allá, desafiando con su productos a herramientas RAD históricamente inexpugnables, incluidas las de Microsoft. Esa decisión ya fue vislumbrada en su momento en las oficinas del gigante de Redmond, que se planteó la posibilidad de terminar por su parte con la tácita convivencia de áreas de ambas compañías.

Por eso, lanzó un sistema de desarrollo de interfaz vectorial que saldría en 2007, con el nombre tentativo de Sparkle, que promete sumergirse en el ámbito de productos orientados al diseño y la multimedia. Es decir que Microsoft no planea competir con Flex, sino directamente meterse con Flash e, incluso, con Director. Hasta algún aventurado ve, en un futuro, a un remozado PowerPoint como una de las patas de la estrategia. Si bien el invicto de Flash lleva ya varios “Flash-killers” en su historial, cabe preguntarse hasta cuándo podrá soportar la embestida comercial de tamaño empresa.

XAML

Al menos cinco años después de que apareció XUL, Microsoft anunció, para su plataforma Longhorn, un lenguaje de marcación que tiene muchos puntos técnicos en común con la tecnología de Mozilla. Más allá de eso, parece que la firma no está del todo conforme con las limitaciones de control del modelo Webforms, y por eso busca mejorar y unificar de algún modo sus interfaces web y de escritorio.

En principio, sin dudas vemos que gran parte del parecido de XAML con XUL es que ambos representan lenguajes declarativos



[Figura 2] Existen varios caminos para llegar a Avalon usando XAML y los lenguajes .NET.



basados en XML. XAML fue desarrollado con el objeto de especificar convenientemente el aspecto de la interfaz de las nuevas aplicaciones de Windows y de proveer una alternativa válida para la creación de esas soluciones. Hasta ese punto, bien podría guardar alguna semejanza con XUL, aunque XAML va bifurcándose más hacia toda la futura plataforma Windows Vista, aportando un canal de comunicación con Avalon, el conjunto de clases encargado de la presentación gráfica y multimedia del futuro sistema operativo. Vale decir que, además de poder llamar directamente a esas clases mediante algún código sostenido en la plataforma .NET, el desarrollador podrá interactuar con Avalon mediante XAML.

C#

```
Button boton = new Button();
boton.Background = Brushes.LightSeaGreen;
boton.FontSize = new FontSize(20, FontSizeType.Point);
boton.Content = "Hola mundo"
```

XAML

```
<Button Background="LightSeaGreen" FontSize="20"
Content="Hola mundo" />
```

Como tercer camino, también podría escribirse C# o VB.NET embebido en XAML, siendo este código manejado al momento de la compilación por sus respectivos gestores. Entendamos bien que con XAML no se busca ni mucho menos reemplazar lenguajes como C# o VB.NET, sino proporcionarles un camino más fácil para utilizar su potencia donde realmente rinden, y dejarles a los diseñadores la tarea de definir la capa de presentación con XAML, como se muestra en la [Figura 2].

AJAX

Antes de definirlo y de pasar a la parte técnica, es útil aclarar dos puntos. El primero es que AJAX no representa una sola tecnología específica, sino un conjunto de ellas que engloba, y define un concepto de trabajo que rompe con el tradicional esquema de aplicación web. Y ni siquiera JavaScript –que quizás es su núcleo central– lo abarca por completo.

El segundo punto es que, si bien AJAX es una modalidad altamente atractiva y usable, no es tan novedosa o revolucionaria como suele verse en ciertos sitios web y blogs de tecnología. Es más, esa tecnología, ahora denominada AJAX, ya venía siendo usada por Microsoft como un objeto Active X. Pero con los años, la modali-

Siempre hay varias tecnologías para elegir, por eso surgen a diario una buena cantidad de APIs destinadas a la generación de GUIs.

dad se extendió en forma más genérica, y cobró mayor notoriedad e impulso a partir de ciertos desarrollos realizados desde su uso en Gmail. AJAX debe su nombre a un artículo escrito por Jesse James Garret, quien lo propuso a fin de darle cierta identidad y hacerlo más extensivo. El nombre es un acrónimo de *Asynchronous JavaScript + XML*, y hace referencia a la creación de una nueva capa de procesamiento dentro del sistema cliente, dirigida por un motor AJAX que va procesando llamadas JavaScript provenientes del cliente, como vemos en la [Figura 3].

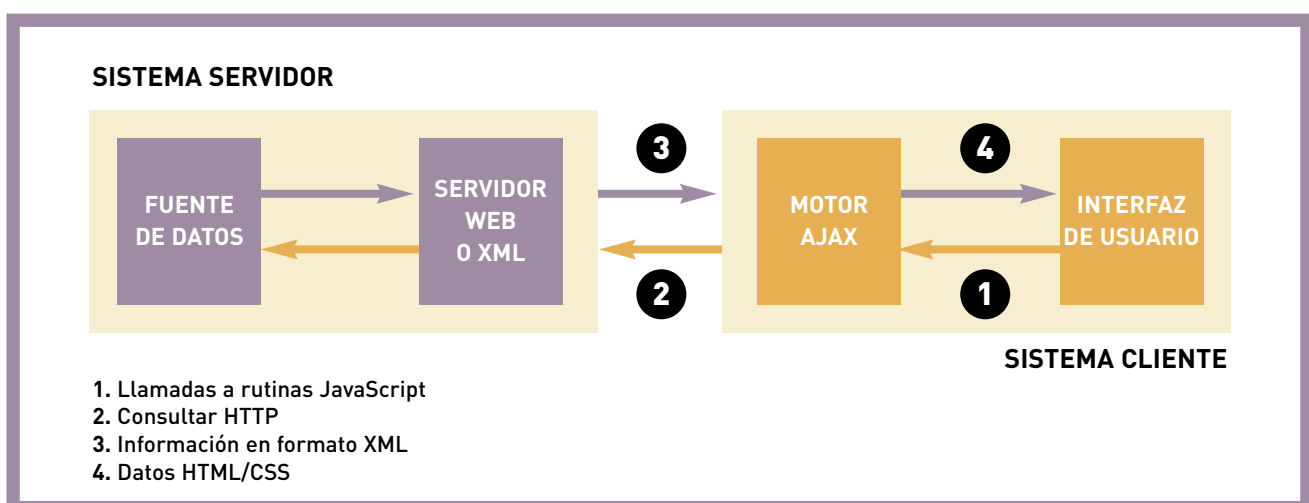
En términos simples, y a modo de ejemplo, podemos citar el caso de un formulario con un combo o *drop down* que llena de elementos su contenido invocando una dirección determinada, de la cual obtiene un archivo xml particular sin tener que recargar la página. Esto se realiza con una serie de rutinas que se ejecutan en el servidor en segundo plano, y que modifican la página de una forma transparente al usuario.

Es destacable aclarar, además, que este útil objeto es compatible tanto en IE como en navegadores Mozilla. En relación con el intercambio y la manipulación de los datos, estos procesos, generalmente, recaen sobre los estándares de XML y XSLT.

Conclusiones

Por último, un consejo: evalúen bien todas las tecnologías y no se definan por ninguna en particular. Todas están aún en franco crecimiento y se concentran, por ahora, sobre plataformas y necesidades desiguales. Por eso, es fundamental no perderle el tren a esta auténtica revolución en Internet. ●

Cristian Hernán Gaitano Ornia es Analista de Sistemas. Se especializa en el área de aplicaciones web y, actualmente, se encuentra trabajando de manera independiente para empresas locales y de los Estados Unidos.



[Figura 3] Éste es el proceso de transporte http mediante AJAX, desde el sistema cliente hasta el servidor de datos.