



ASP.NET

Hoy en día, resulta bastante común implementar la interfaz de una aplicación utilizando páginas web en vez de las ventanas y los controles específicos de un sistema operativo concreto. En lugar de escribir una aplicación para un sistema operativo concreto, como puede ser Windows, en muchas situaciones es preferible crear aplicaciones web a las que se accede a través de Internet.

Se denominan aplicaciones web a aquellas aplicaciones cuya interfaz se construye a partir de páginas web. Las páginas web no son más que ficheros de texto en un formato estándar denominado HTML [*HyperText Markup Language*]. Estos ficheros se almacenan en un servidor web al cual se accede utilizando el protocolo HTTP [*HyperText Transfer Protocol*], uno de los protocolos de Internet. Para utilizar una aplicación web desde una máquina concreta, basta con tener instalado un navegador web en esa máquina, ya sea éste el Internet Explorer de Microsoft, el Netscape Navigator o cualquier otro navegador. Desde la máquina cliente, donde se ejecuta el navegador, se accede a través de la red al servidor web donde está alojada la aplicación y, de esa forma, se puede utilizar la aplicación sin que el usuario tenga que instalarla previamente en su máquina.

Si las páginas que forman la interfaz de nuestra aplicación las construimos utilizando única y exclusivamente HTML estándar, podemos conseguir que nuestra aplicación funcione sobre prácticamente cualquier plataforma, siempre y cuando dispongamos de un navegador web para el sistema operativo instalado en la máquina desde la que queramos acceder a la aplicación. Una interfaz construida de esta manera nos permite olvidarnos de los detalles específicos de los diferentes entornos gráficos existentes. Dichos entornos gráficos suelen

depender del sistema operativo que utilicemos. Por ejemplo, en Windows utilizaremos las ventanas típicas a las que ya estamos acostumbrados, mientras que en UNIX emplearíamos el sistema X Windows y en Linux podríamos utilizar gestores de ventanas como KDE o Gnome.

Como no podía ser de otra forma, el aumento de la portabilidad de nuestra aplicación lleva consigo ciertas limitaciones en lo que respecta a su usabilidad. Esto se debe a que la interacción del usuario con el ordenador a través de los formularios HTML empleados en las páginas web está mucho más limitada que en el caso de que podamos aprovechar la funcionalidad que ofrecen los controles específicos existentes en la mayoría de los entornos gráficos basados en ventanas.

En el desarrollo de la interfaz de nuestra aplicación podríamos evitar, al menos en parte, la solución de compromiso que supone utilizar formularios HTML para conseguir aplicaciones portables. Existen bibliotecas portables para el desarrollo de interfaces gráficas, como es el caso de *SWING* en la plataforma Java de Sun Microsystems, o de productos como *Qt* (Troll Tech, <http://www.troll.no>). En el caso de la plataforma Java, ésta se diseñó como una máquina virtual para que pudieran ejecutarse las mismas aplicaciones en distintos sistemas operativos sin tener ni siquiera que recompilar el código. Por su parte, las bibliotecas del estilo de *Qt* ofrecen conjuntos de componentes con implementaciones de los mismos para distintos entornos basados en ventanas. En una de estas bibliotecas de componentes (o tal vez deberíamos decir bibliotecas de interfaces), las distintas implementaciones de un componente particular tienen los mismos interfaces para todas y cada una de las plataformas sobre las que esté implementada la biblioteca. De este modo, si deseamos portar nuestras aplicaciones de un sistema a otro sólo tendremos que recompilar nuestra aplicación utilizando la versión de la biblioteca adecuada para el sistema operativo sobre el que queremos ejecutar la aplicación, sin tener que modificar ni una sola línea en nuestro código fuente. Dicho código fuente, obviamente, lo tendremos que escribir evitando cualquier uso de recursos específicos de una plataforma (algo, de por sí, extremadamente difícil).

Aunque pueda parecer que soluciones como las descritas en el párrafo anterior resuelven nuestros problemas a la hora de desarrollar aplicaciones portables sin limitaciones en cuanto a la funcionalidad de su interfaz, en realidad se limitan a resolver una pequeña parte del problema que supone instalar y mantener un sistema funcionando. De hecho, la etapa del ciclo de vida del software que más recursos consume es la fase de mantenimiento.

El mantenimiento del software consume del cuarenta al ochenta por ciento del coste de un sistema y, probablemente, sea la etapa más importante del ciclo de vida del software (aparte de ser la más olvidada). No debemos pensar que el mantenimiento sólo consiste en reparar aquello que deja de funcionar por un defecto de fabricación o por el desgaste asociado al paso del tiempo. La naturaleza del software hace que nunca se rompa por desgaste. De todos los gastos asociados al mantenimiento del software (que viene a suponer un 60% de su coste total), un 60% corresponde a la realización de mejoras, mientras que sólo un 17% está directamente asociado a la corrección de defectos. Por tanto, el 60% del 60% del coste total del software está asociado a la realización de actualizaciones. Si nuestra aplicación ha de funcionar en muchos puestos diferentes, cada pequeña actualización que realicemos supondrá un esfuerzo considerable a la hora de su distribución e instalación. Y si nuestro sistema tiene

éxito, habrá muchas de esas actualizaciones.

Teniendo los hechos anteriores en cuenta, resulta evidente el atractivo que tiene la implementación de aplicaciones web, puesto que se elimina el problema de la distribución de actualizaciones cada vez que modificamos un sistema. Cuando tengamos que realizar una actualización, nos bastará con modificar la configuración del servidor que da acceso a nuestra aplicación web para que todos los clientes dispongan automáticamente de la versión más reciente de la aplicación. Además, el usuario podrá acceder a nuestra aplicación desde cualquier plataforma y nosotros podremos implementarla utilizando todos los recursos disponibles en la plataforma que elijamos para albergarla, sin restricciones innecesarias (si bien esto no quiere decir que no aspiremos a la construcción de una aplicación flexible que nos facilite el trabajo si algún día tenemos que cambiar de servidor de aplicaciones).

Dada la importancia actual de las aplicaciones web, el siguiente capítulo ofrece una introducción al desarrollo de interfaces web para ayudarnos a comprender en qué consisten y cómo se construyen las aplicaciones de este tipo. En los capítulos posteriores de esta parte del libro veremos con más profundidad algunos de los detalles de la construcción de interfaces web con ASP.NET, la tecnología de la plataforma .NET que nos permite desarrollar aplicaciones web.

Referencias

Los datos que hemos mencionado relacionados con el mantenimiento del software se analizan en el libro de Robert L. Glass titulado "*Facts and fallacies of Software Engineering*" (Addison-Wesley, 2003, ISBN 0-321-11742-5). Este libro es una obra interesante en la que se puede consultar información adicional acerca de cuestiones fundamentales que a menudo se nos olvidan cuando estamos enfrascados en el desarrollo de software.

