

Sistema Web basado en ASP y XML para la Gestión Presupuestaria de un Centro Universitario

Silvia Conde Rey

E.S. de Ingeniería Informática
Universidade de Vigo
32004 - Ourense
silvia.conde@terra.es

Eva Dafonte Pérez

E.S. de Ingeniería Informática
Universidade de Vigo
32004 - Ourense
evitadaf@terra.es

Enrique Barreiro Alonso

E.S. de Ingeniería Informática
Universidade de Vigo
32004 - Ourense
enrique@uvigo.es

Resumen

La aplicación que aquí se presenta ha sido desarrollada como Proyecto Fin de Carrera (PFC) en la titulación de Ingeniería Técnica en Informática de Gestión de la Escuela Superior de Ingeniería Informática (ESII) de la Universidad de Vigo. Representa un ejemplo de las ventajas que, tanto para el alumno como para los Centros y Departamentos aporta el enfocar el esfuerzo empleado en estos PFCs hacia el desarrollo y mejora de los servicios ofrecidos a través de Internet o las Intranets universitarias.

Esta aplicación, que está en explotación actualmente en la ESII, permite gestionar, a través de un entorno web, el presupuesto anual adjudicado a un centro universitario, y ha sido realizado utilizando técnicas y lenguajes cada vez más extendidos y demandados por el mercado. El objetivo del PFC era investigar la utilidad de la programación XML en el desarrollo de aplicaciones de gestión y la conveniencia de combinar el lenguaje XML y la programación ASP en aplicaciones de esta naturaleza. El modelado y diseño de la aplicación se ha realizado con una extensión de UML para aplicaciones Web.

1. Introducción.

El desarrollo de los PFC, que implica un esfuerzo considerable por parte del alumno al finalizar las asignaturas de su titulación, debe ser aprovechado en lo posible para aproximar al alumno a los problemas reales del desarrollo de

software, con requisitos establecidos por usuarios que realmente van a utilizar el sistema. Esta problemática, a pesar de cursar varias asignaturas relacionadas con la Ingeniería del Software, queda bastante desdibujada a lo largo de los estudios, pues, aunque se realicen proyectos completos en las prácticas de las asignaturas, la duración de éstas y el número de alumnos no permiten que el profesor pueda "ejercer" de cliente de un modo satisfactorio.

Por otra parte, y aunque cada vez más empresas se muestran interesadas en colaborar con la Universidad en proyectos de I+D, no siempre resultan adecuados y suficientes para acoger a todos los alumnos que realizan su PFC cada año. Así, desde hace unos años, el Departamento de Informática dirige PFCs que contribuyen a aumentar y mejorar los servicios que se ofrecen a través de los servidores Web del propio Departamento y de la Escuela, tanto al exterior como a través de sus respectivas Intranets, lo que redundará en una mejora de la calidad del servicio de aquellos.

Además, el desarrollo de este tipo de aplicaciones permite, tanto a alumnos como a profesores, adquirir conocimiento y experiencia en tecnologías que, normalmente, son muy recientes y que no siempre se incorporan a los planes de estudio con la rapidez que sería deseable. El alumno, así, puede incorporarse al mercado laboral con un bagaje de conocimientos especialmente atractivo y valorado por las empresas.

Con el PFC aquí presentado se planteaba, como objetivo genérico, la investigación de la utilidad de la programación de aplicaciones de gestión mediante XML (eXtended Markup Language), un lenguaje de etiquetas emergente y que representa un salto cualitativo en el desarrollo de sistemas basados en web

(principalmente aplicaciones de comercio electrónico y de gestión documental), así como su combinación con la programación ASP (Active Server Pages) en aplicaciones de esta naturaleza. Al mismo tiempo, se pretendía estudiar las ventajas e inconvenientes del uso de UML (Unified Modelling Language) en el modelado y diseño de aplicaciones de gestión basadas en tecnologías web.

Los conocimientos de ASP, XML y UML adquiridos en las asignaturas optativas *Programación Avanzada*, *Diseño de Interfaces de Usuario* y *Metodologías Avanzadas de Desarrollo de Software* se ampliaron y aplicaron en el desarrollo de un sistema de gestión económica de un centro universitario.

2. Requisitos de la aplicación.

El sistema permite gestionar, desde un entorno web, el presupuesto anual adjudicado a un centro universitario. Este presupuesto está repartido en distintas partidas presupuestarias (libros, equipamiento, etc.) y algunas de ellas, a su vez, son distribuidas entre las áreas de conocimiento del centro. Se puede conocer el estado de cada partida, consultar los gastos realizados por un área, elaborar informes anuales... La aplicación también permite gestionar los distintos departamentos del centro, así como las áreas de conocimiento que los componen y los profesores con docencia en dicho centro.

Las funciones que debía incorporar la aplicación (ver Diagrama de Casos de Uso en Figura 1) fueron establecidas por los usuarios de ésta (la Dirección y el responsable de Asuntos Económicos del Centro) y se resumen a continuación:

- Iniciar un ejercicio económico.
- Realizar un nuevo reparto presupuestario correspondiente al ejercicio económico abierto con los presupuestos ofertados.
- Actualizar el reparto existente con la llegada de partidas extraordinarias.
- Crear nuevas partidas en el sistema cuando las existentes no son suficientes para recoger todos los conceptos en los que se divide el presupuesto, así como la eliminación de aquellas que hubiesen quedado obsoletas.

- Permitir la realización de actualizaciones sobre el reparto presupuestario entre las áreas de conocimiento de un centro universitario. Estas actualizaciones son debidas a los gastos correspondientes a las diversas partidas que a lo largo del ejercicio económico irán solicitando las distintas áreas de conocimiento del centro. La cancelación de un gasto también dará lugar a una operación de actualización de saldos en el sistema.
- Gestionar los distintos departamentos del centro, así como las áreas de conocimiento que los componen y los profesores con docencia en dicho centro, permitiendo la creación, modificación de los datos relativos a los mismos y la supresión de cualquiera de ellos.
- Realizar consultas referentes al saldo actual de cualquiera de las partidas de un área concreta, así como consultas referentes a los últimos movimientos realizados sobre una determinada partida, además de todas las transacciones realizadas a lo largo del año en el que transcurre el ejercicio económico.

3. Descripción técnica.

3.1. Solución adoptada.

La solución adoptada se basa en la utilización de tecnologías propias de Internet, que permiten el desarrollo de aplicaciones distribuidas basadas en el modelo cliente/servidor.

Las aplicaciones web suponen un importante cambio de enfoque con respecto al desarrollo de aplicaciones tradicionales. Su principal característica consiste en que la comunicación con el usuario se establece utilizando páginas HTML, que se pueden visualizar desde un navegador que se esté ejecutando en cualquier ordenador conectado a la red. Otra característica importante, consiste en que el código de la aplicación se puede ejecutar en el cliente, en el servidor o distribuirse entre ambos. Las aplicaciones web presentan las siguientes ventajas: (1) desarrollo evolutivo y relativamente rápido, (2) accesibilidad, ya que cualquier usuario autorizado puede utilizarla si dispone de conexión a Internet, (3) soporte

multiplataforma por medio de exploradores web, (4) facilidad de manejo y (5) fácil mantenimiento de las aplicaciones.

Se puede decir, en términos generales, que las aplicaciones web tienen dos componentes principales: la parte del cliente (front-end) y la parte del servidor (back-end). El lado del cliente se puede crear utilizando un lenguaje que funcione dentro de las limitaciones de la computadora y el explorador del usuario. La interfaz del cliente se crea con HTML y scripts. La parte del servidor de esta aplicación debía trabajar bajo Windows NT Server y acceder a la información mediante el servidor de páginas web Internet Information Server, de Microsoft.

La elección de la programación ASP (Active Server Pages) para el desarrollo de esta aplicación demostró ser adecuada, debido a su potencia, combinada con una gran facilidad de aprendizaje. Esta tecnología, propuesta por Microsoft, incluye tanto HTML como sentencias de programa para las que se pueden utilizar diferentes lenguajes de scripts, (como Visual Basic Script o JavaScript). Una de las ventajas de ASP es que permite crear objetos ActiveX que se ejecutan en el servidor de modo que soporta acciones como acceso a base de datos mediante el modelo ADO (ActiveX Data Object) a través de ODBC consiguiendo una total integración con un gestor de bases de datos fiable y potente como es SQL Server 7. ODBC proporciona una interfaz que permite a la aplicación acceder a la información contenida en las bases de datos a través de sus distintos controladores, independientemente del sistema gestor de bases de datos utilizado.

El módulo de control de ASP necesita ser instalado en el servidor web Internet Information Server (en las últimas versiones ya viene integrado), que a su vez sólo puede ser instalado en sistemas operativos Windows. Esta característica permite que todos estos sistemas se complementen entre sí, para conseguir una plataforma adecuada en el desarrollo de aplicaciones web. No obstante, existen extensiones ASP para otro tipo de servidores web y otras plataformas (por ejemplo, en servidores Apache corriendo bajo Linux)

Microsoft SQL Server permite construir sitios web potentes y dinámicos. Con Microsoft Internet Information Server (IIS) se tiene acceso

a una plataforma de desarrollo web que cumple todas las normas actuales del sector. Se puede emplear una gran cantidad de lenguajes de programación para acceder a la información a través de IIS.

3.2. Análisis y diseño.

Para el análisis y diseño del sistema se utilizó el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) [1]. Se ha comprobado que para el diseño de aplicaciones Web, UML presenta una serie de limitaciones que le impiden representar algunas de las cuestiones importantes que se reflejan en una aplicación de estas características. Para solucionar esta limitación es imprescindible crear una serie de notaciones y métodos de desarrollo y por ello se ha recurrido a una extensión de UML (*Web Application Extension*) [2], que proporciona mecanismos que permiten añadir una semántica adicional a la notación y nuevos bloques de construcción y propiedades con restricciones para permitir el modelado de elementos específicos en una aplicación web. Los estereotipos y restricciones se aplican a ciertos componentes particulares de los sistemas web (por ejemplo, formularios, páginas ASP, librerías script o páginas de servidor), y permiten representarlos en el mismo modelo y en los mismos diagramas que describen el resto del sistema.

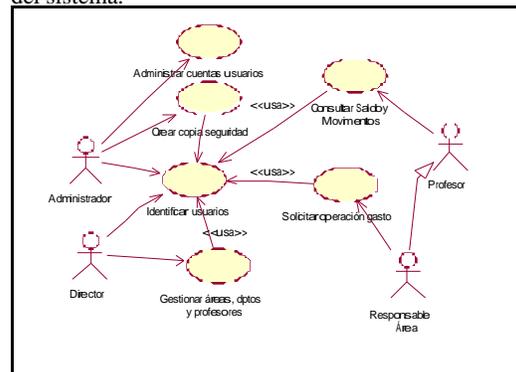


Figura 1. Diagrama parcial de Casos de Uso.

A continuación se citan brevemente los diagramas de UML que se han utilizado en el modelado y el diseño del sistema:

- *Diagramas de casos de uso.* Utilizados para modelar los aspectos dinámicos del sistema. Se complementan con una descripción textual para cada caso de uso identificado. En la Figura 1 se muestra una parte del diagrama de casos de uso de la aplicación..
- *Diagrama de clases.* Expresa de manera general la estructura estática del sistema, en términos de clases y sus relaciones (Figura 2).

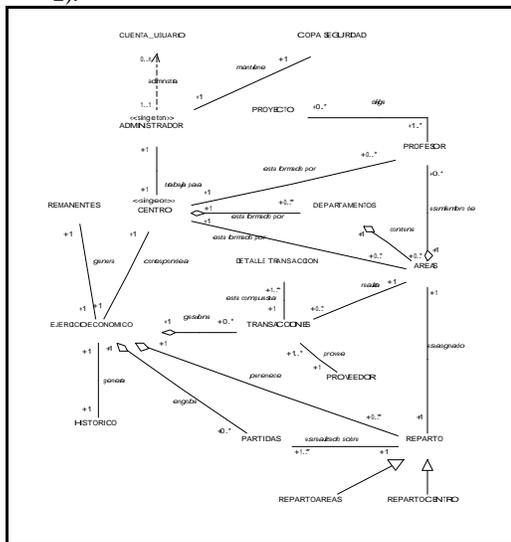


Figura 2. Diagrama de clases

- *Diagramas de secuencia.* Muestran un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos, enfatizando la cronología de los envíos de estos mensajes. Por las especiales características de una aplicación web de gestión, la extensión de UML mencionada ha resultado ser de gran utilidad, pues permite durante el modelado el comportamiento dinámico de una aplicación de estas características.
- *Diagramas de estados.* Describen el comportamiento del sistema, anotando de forma individual los posibles estados que los objetos pueden tener y como cambian estos como resultado de los eventos que reciben.

- *Diagramas de actividades.* Desarrollados para alcanzar una mayor comprensión del funcionamiento del sistema.
- *Diagramas de componentes.* Modelan aspectos físicos del sistema mostrando la organización y dependencias entre un conjunto de componentes. En la Figura 3 se puede observar un diagrama de componentes que representa la estructura utilizada en la administración de usuarios del sistema, y en el que se han aplicado estereotipos para representar páginas ASP y librerías de script, pertenecientes a la extensión de UML.

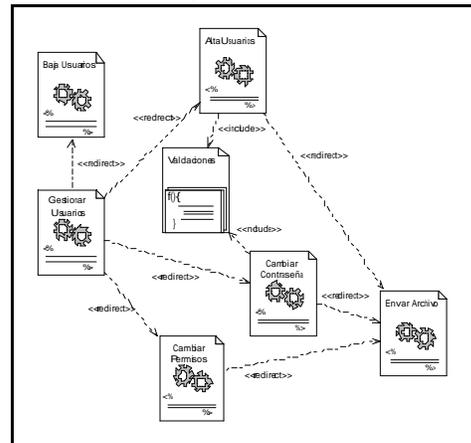


Figura 3. Diagrama de componentes que muestra la estructura de la administración de usuarios.

4. Consideraciones sobre la tecnología utilizada.

4.1. El enfoque de objetos en los desarrollos web y UML.

Una de las grandes ventajas de adoptar un enfoque orientado a objetos en el desarrollo de este tipo de sistemas es que resultan más flexibles de adaptar, modificar e incluir nuevas funcionalidades, algo bastante habitual en cualquier sistema de información, y, dada su novedad y la evolución de la tecnología, aún más en los sistemas basados en tecnologías web. Aunque en un primer momento la adaptación al enfoque orientado a objetos no resulta sencilla (seguramente como consecuencia de que el

estudio de la programación, hasta hace un par de años, se basaba en un enfoque eminentemente estructurado), posteriormente este acaba resultando más sencillo, intuitivo y atractivo. El desarrollo orientado a objetos también tiene algunos riesgos (costes y eficacia) pero la experiencia indica que los beneficios superan con mucho los riesgos.

El proyecto que aquí se presenta se ha desarrollado en su totalidad en poco más de tres meses, y sin duda una de las claves de esta rapidez del desarrollo ha sido la gran expresividad del enfoque de objetos en general y de UML en particular. Como se dijo anteriormente, la utilización de UML en este tipo de aplicaciones lleva asociadas una serie de limitaciones en cuanto a su expresividad, por lo que la extensión de UML para aplicaciones web ha permitido una mejor comunicación con los usuarios finales. La facilidad en la definición de las distintas funcionalidades del sistema, los diferentes flujos de trabajo de la aplicación, usuarios, formularios, etc, permitieron abordar el desarrollo de un modo realmente evolutivo e incremental.

4.2. ASP y las aplicaciones cliente/servidor.

En las aplicaciones cliente/servidor basadas en la Web la información se encuentra almacenada en un servidor. Para ver o manipular los datos del servidor se emplea una aplicación cliente y los datos pueden estar almacenados en una base de datos. El principal problema que presenta la adopción de la tecnología Web para el desarrollo de aplicaciones distribuidas es la seguridad. Se necesita el uso de medidas que garanticen la utilización de la aplicación únicamente por usuarios autorizados, sobre todo si las operaciones que se pueden realizar son críticas para el sistema. Se ha comprobado que la tecnología ASP permite llevar a cabo, de un modo bastante satisfactorio, la gestión de usuarios de un sistema.

Las páginas ASP [5] constituyen un entorno de scripts basados en servidor que se utilizan para desarrollar aplicaciones Web dinámicas e interactivas. La ejecución de scripts en el servidor, en lugar de en el cliente, permite emplear toda la potencia del servidor web. Estos scripts pueden ser programados en cualquier

lenguaje de scripts para el que se pueda proporcionar el apropiado motor de scripts, como VBScript y JavaScript. Aunque VBScript permite interactuar con objetos ActiveX, lo que ofrece una ventaja significativa frente a la programación con JavaScript, tiene la desventaja de que muchas plataformas no lo soportan. Por otro lado, JavaScript proporciona un conjunto de objetos predefinidos de gran utilidad y sirve de base para incorporar otros elementos tecnológicos como son ActiveX, XML o HTML dinámico razón principal por la que se ha utilizado mayoritariamente JavaScript dentro de las páginas ASP para conseguir un funcionamiento óptimo de la aplicación [3].

4.3. XML y ASP.

La aplicación principal de XML [4] en este sistema es la edición de información en la Web a partir de bases de datos. Por ejemplo, para la consulta de información referente al saldo actual de las partidas presupuestarias a través de la Web se establece una conexión entre el servidor Web y la base de datos de la aplicación de modo que, a medida que se vayan realizando gastos o devoluciones sobre dicha partida, estos cambios aparezcan automáticamente al visitar las páginas Web. Aunque en este proyecto XML se ha utilizado básicamente para la estructuración y presentación digital de los documentos, la generación de documentos XML a través de las páginas ASP resulta sencilla (en la Figura 4 se puede ver un ejemplo de código), lo que abre una gran cantidad de posibilidades para el desarrollo de sistemas de información empresarial.

No obstante, se ha podido comprobar que algunos de los elementos de HTML no son compatibles con XML como por ejemplo los caracteres especiales. Cuando se incluyen caracteres especiales de HTML o palabras acentuadas dentro de las etiquetas de XML se puede producir un funcionamiento anómalo de la página Web que llama al documento XML generado.

Debido a la relativa juventud de XML cabe esperar mejoras en los próximos años. Sin embargo, ya está teniendo un gran impacto en el desarrollo de aplicaciones de negocio electrónico basadas en tecnologías de Internet.

```

(...)
'Comienzo del documento XML que se construye dentro de
las páginas ASP
%>
<XML ID=xmlDetalle>
<?xml version="1.0?">
<TRANSACTION>
<%
Do while NOT rs.EOF
%>
<DETALLE>
<CODIGO><%=rs("Codigo_Detalle")%></CODIGO>
<CONCEPTO><%=rs("Concepto")%></CONCEPTO>
<CANTIDAD><%=rs("Cantidad")%></CANTIDAD>
<% if unidad = "euros" then %>
<PRECIO><%=rs("Precio")&"
€."%></PRECIO>
<% else %>
<PRECIO><%=rs("Precio")&"
Pts."%></PRECIO>
<% end if %>
<% importe=importe+(rs("Cantidad")*rs("Precio"))
total=cambiar_cantidad(importe)
sql="UPDATE Transaction SET
Importe_Total="&total&"where
N_Hoja_Pedido="&numero&"
cn.Execute(sql)
%>
</DETALLE>
<% rs.MoveNext
Loop
%>
</TRANSACTION>
</XML>
<%
'Fin del documento XML
%>
(...)
```

Figura 4. Ejemplo de generación de documento XML desde una página ASP

5. Conclusiones técnicas.

La utilización de la tecnología ASP combinada con XML ha demostrado ser de gran utilidad cuando se aborda el desarrollo de aplicaciones de gestión basadas en tecnologías web. Sin embargo es importante considerar el factor novedad por lo que de riesgo puede incorporar a los proyectos. Tanto las técnicas de modelado y diseño de aplicaciones como los propios lenguajes son relativamente recientes y poco probados. Cualquier empresa que pretenda ubicar sus procesos de negocio en la web (intranets, extranets,...) debe apostar por tecnologías probadas, adquiriendo poco a poco la experiencia necesaria mediante el desarrollo

de aplicaciones y servicios cada vez más complejos.

6. Conclusiones docentes.

De este y otros proyectos dirigidos en los últimos cursos en la ESII de la Universidad de Vigo se ha podido concluir que resulta especialmente interesante para la formación del alumno y su proyección en el mercado laboral el aprovechamiento de los PFC para aprender lenguajes y técnicas de desarrollo especialmente valoradas en la actualidad. El escaso periodo de desarrollo empleado en este proyecto viene determinado, en gran medida, por los conocimientos que sobre estas técnicas se estudiaron en las asignaturas optativas *Programación Avanzada, Diseño de Interfaces de Usuario y Metodologías Avanzadas de Desarrollo de Software*. Parece, pues, acertada la política de introducir en el plan de estudios dos o tres materias optativas relativamente genéricas con descriptores flexibles y pensadas para incorporar técnicas y lenguajes de reciente aparición, y cuya incorporación a la docencia obligatoria y troncal suele resultar más lenta

Por otro lado ha quedado patente la gran utilidad de aprovechar el propio centro universitario como campo de desarrollo de los PFC, pues permite, por un lado, que los alumnos trabajen con un cliente real y en un entorno cercano, al tiempo que se consigue aprovechar en beneficio de la propia Universidad los esfuerzos empleados en la dirección y desarrollo de los PFC.

Referencias

- [1] G. Booch, J. Rumbaugh, I. Jacobson. *El Lenguaje Unificado de Modelado*. Addison Wesley, 2000.
- [2] J. Conallen. *Building Web Applications with UML*. Adison Wesley, 2000.
- [3] J. Bobadilla Sancho. *Superutilidades para Webmasters*. McGrawHill, 1999.
- [4] D. Martin et al. *Professional XML*. Wrox, 2000.
- [5] M. Wilson, T. Wilson. *XML Programming with VB and ASP*. Manning, 1999.