

Un Sistema de Información de Libre Distribución: la Herramienta OpenONG para la Gestión de ONGs

Manuel R. Cabello, José G. Moreno-Torres, Luis González,
Daniel López, José F. Mantas, Ricardo Palma,
Víctor Terrón, Javier Toral, Pablo Villacorta
Alumnos de la E.T.S. de Ingenierías Informática y de Telecomunicación,
Universidad de Granada.
C/ Periodista Daniel Saucedo Aranda, s/n. 18071 Granada, España
{ridick, josegmt, nerit, gothmog, jfranc, odracir,
quintan, jtn85, olbapjo}@correo.ugr.es

Resumen. Se ha modelado, diseñado e implementado un software libre multi-plataforma y orientado a objetos para la gestión de los aspectos más habituales de las Organizaciones No Gubernamentales (ONGs). La solución propuesta posee características técnicas notables como un diseño muy robusto y estructurado aplicando el modelo MVC, con alta reutilización dentro de la aplicación, además de una gran homogeneidad visual en una interfaz de usuario totalmente intuitiva y amigable. El software se completa con el desarrollo de un Sitio Web que se genera automáticamente a partir de los datos almacenados por la ONG.

Palabras Clave: OpenONG, ONG, Software Libre, Software Multiplataforma, Modelo-Vista-Controlador.

1 Introducción

El auge del Software Libre como alternativa de calidad al software propietario tradicional ha llevado al fomento del desarrollo de herramientas de este tipo. Nuestra solución pretende aplicar las nuevas tecnologías a la gestión y organización de una ONG, creando una aplicación intuitiva y de fácil manejo para usuarios que no siempre están familiarizados con el mundo de la informática. Queremos presentar una aplicación alternativa a gONG [1], que gestione los miembros, proyectos y eventos de una ONG.

2 Requerimientos que cumple la solución

La funcionalidad que ofrece se puede dividir en varios apartados. Por un lado, la gestión de personal de la ONG. Se distinguen socios, voluntarios, cooperantes y trabajadores de empresas externas contratadas por la ONG. Por otro lado, la gestión de proyectos desarrollados por la ONG. Cada proyecto a su vez podrá estar compuesto por una o más tareas, a las que podremos asociar miembros de la ONG que colaboran en las mismas. También permite asociar imágenes a un proyecto me-

diante un servidor de FTP, y de clasificar un proyecto en una o varias categorías definidas por el propio usuario. En tercer lugar, la gestión de eventos convocados por la ONG. Se distinguirán eventos públicos y privados, siendo estos últimos visibles en el Sitio Web sólo para socios, previa autenticación en la web. Además, se realiza control de acceso al Sistema que permite que la funcionalidad a la que se tiene acceso varíe en función de los privilegios del usuario en la ONG. Adicionalmente, se lleva a cabo la generación automática de un Sitio Web de la ONG. Finalmente, todo lo anterior se ha implementado de manera que la aplicación sea multiplataforma (Windows y GNU/Linux). Se suministra un instalador sencillo para cada plataforma.

4 Características técnicas destacadas de OpenONG

4.1 Diseño de objetos: aplicación del patrón Modelo-Vista-Controlador

Después del análisis de requerimientos, concluimos que el patrón arquitectónico Modelo-Vista-Controlador (en adelante, MVC [2]) era el que más se adecuaba a los requisitos que deseábamos alcanzar. El diagrama de clases se ha diseñado, por tanto, siguiendo el patrón MVC. Para cada modelo, debería existir un controlador y una vista. El acceso identificado de la aplicación representa el modelo MVC en su más pura esencia: una clase modelo, otra vista y otra controladora para gestionar el acceso a la aplicación (Fig. 1). Sin embargo, dado que seguir este modelo para cada una de las posibles vistas del Sistema sería largo y tedioso, se decidió simplificar el diseño general: se redujeron la mayoría de los posibles controladores y vistas, lo cual resultó en un diseño general que, si bien no se corresponde de forma exacta con la filosofía MVC, sí la logra de forma general y conserva sus virtudes principales.

Cada clase controladora tiene una serie de *puertas de acceso* que permiten a otras clases vista y a otros controladores comunicarse con ella. Las puertas de acceso restringen la visión que se tiene de una clase determinada. La razón es sencilla: si una vista tiene que comunicarse con un controlador, no tiene sentido que dicha vista tenga una percepción de toda la funcionalidad del controlador, sino sólo una visión parcial, a saber: la necesaria para la funcionalidad que esa vista requiere de dicho controlador. Análogamente para el caso de la comunicación entre controladores, no tiene sentido que un controlador secundario que depende de uno principal pueda acceder a toda la funcionalidad del principal, sino sólo a aquello que le es de interés de éste.

Esta peculiaridad de nuestro diseño ha sido lograda mediante la creación de interfaces, llamadas *Acción*. Cuando una clase A (ya sea controladora o vista) necesita comunicarse con una clase controladora B, la clase B implementará una interfaz *Acción* que represente la visión que A necesita de B. La clase A no contendrá por tanto una referencia a la clase controladora B sino a la interfaz *Acción* que B implementa. Así, el diseño gana en claridad y legibilidad: cuando una clase A se comunica con B a través de una interfaz *Acción*, aislamos el resto de funciones de B que no son de interés para A.

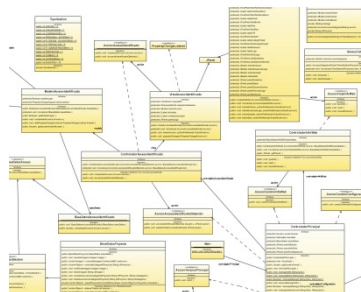


Fig. 1. Fragmento del diagrama de clases del acceso identificado.

4.2 Alta homogeneidad visual, fácil manejo y navegabilidad intuitiva

El diseño de la interfaz de usuario se realizó teniendo en cuenta que la aplicación sea fácil de manejar para personal no especializado. Por ello, la funcionalidad análoga para la gestión de aspectos diferentes está siempre dispuesta visualmente de la misma manera, a fin de que el usuario aprenda rápidamente a manejar la aplicación.

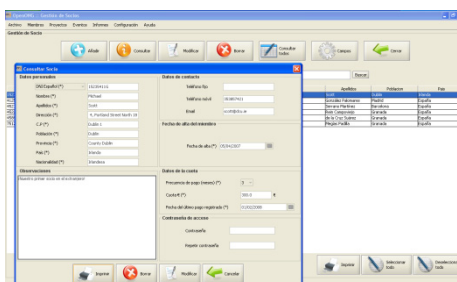


Fig. 2. Menú de gestión de socios. La apariencia visual de la gestión del resto de apartados de la ONG es prácticamente idéntica.

También se diseñó de manera que la navegabilidad entre menús relacionados sea lo más rápida y directa posible, sin obligar a rodeos innecesarios. Por ejemplo, se puede borrar un miembro (o un proyecto, o una tarea, ya que el tratamiento es análogo) desde la tabla donde se listan, desde la pantalla de consulta de los datos de un miembro concreto, o desde la de modificación de dichos datos, tal como muestra la Fig. 2.

4.3 Alta escalabilidad, reusabilidad del diseño y reusabilidad de código

La aplicación del patrón MVC también ha permitido que el Sistema sea muy fácilmente extensible. Por tanto, es posible adaptarlo fácilmente a las necesidades de otras ONGs si no están ya contempladas, o añadirle funcionalidad adicional, lo cual pone a nuestra solución en una posición ventajosa respecto a otros sistemas existentes y menos extensibles. Esto también es consecuencia de la alta reutilización de paneles gráficos y de código existente dentro de la propia aplicación.

Es más, el diseño elaborado (y consecuentemente, el código generado) se puede adaptar incluso a otras aplicaciones de gestión similares, muy comunes hoy en día. Casi cualquier aplicación donde sea necesario gestionar miembros y algunos aspectos relacionados puede construirse a partir de nuestro diseño, minimizando el esfuerzo de desarrollo y permitiendo a los nuevos desarrolladores centrarse en las peculiaridades de su problema para adaptar nuestra solución o añadirle características nuevas.

4.4 Sitio Web generado automáticamente

El Sitio Web creado tiene principalmente fines promocionales: muestra información de la ONG, sus estatutos, eventos y proyectos. El contenido del Sitio se lee dinámicamente de la BD, lo cual implica que no es necesario actualizarlo manualmente. Por ejemplo, si desde la aplicación se añade una nueva foto a un proyecto, el cambio se reflejará automáticamente en la Web.

4.5 Otros detalles de implementación

Otros aspecto importante es la posibilidad de configurar la dirección del servidor de BD y de FTP (donde se almacenan las imágenes), así como el login y contraseña utilizados para el acceso a cada uno de ellos. Esto se guarda en un archivo de configuración cifrado mediante TripleDES. Todas las contraseñas se almacenan en la BD cifradas con el algoritmo SHA-1. Por otro lado, el Sistema incorpora un menú de ayuda integrada, y permite la obtención de listados en papel de miembros, proyectos y tareas, y la generación de informes con la información relevante entre 2 fechas dadas.

5 Conclusiones y trabajos futuros

Hemos presentado un prototipo de un Sistema de Información Libre multiplataforma y orientado a objetos para la gestión de una ONG. Su diseño fácilmente escalable, su homogeneidad visual, su manejo intuitivo para el usuario y otros detalles de implementación lo hacen valioso para su utilización en entornos reales. Como posibles mejoras futuras, se proponen las siguientes: control de concurrencia para varias instancias simultáneas de la propia aplicación contra una misma Base de Datos, adaptación a otras plataformas, e incremento de la configurabilidad y seguridad del Sitio Web.

Referencias

1. CENATIC: Análisis funcional del Sistema de Gestión de Proyectos gONG. http://www.cenatic.es/proyectos/blogs/media/blogs/a/Analisis_Funcional_GONG_0106.pdf. Accedido el 15 de Junio de 2008
2. Larman, C.: *UML y Patrones*. 2ª Edición. Pearson (2003)