

DESARROLLO DE SISTEMAS BASADOS EN ESTÁNDARES Y CÓDIGO OPEN SOURCE CASO DE ESTUDIO: CALENDARIO DISTRIBUIDO

Ortiz, P.; Ing. Abad, C.

Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral
portiz@fiec.espol.edu.ec, cabad@fiec.espol.edu.ec
Teléfonos: (+593 4) 2269 - 300, (+593 4) 2269 - 268.
Fax: (+593 4) 2853-078

Resumen

El objetivo de este artículo es proponer el uso del código abierto y estándares como una de las mejores prácticas actuales en el desarrollo de software y sistemas. Se expone el creciente interés global alrededor de estas herramientas, basado en gran medida al fácil acceso al Internet que permite a programadores distribuidos mundialmente la formación de comunidades donde la colaboración habilita la producción de código de alta calidad, con menores defectos y menores vulnerabilidades de seguridad que su contraparte el software propietario y a un costo relativamente bajo.

Se plantea seguidamente la necesidad de emplear estándares abiertos para facilitar la interoperabilidad con el resto de sistemas y aplicaciones.

La aplicación de este modelo es enfocado en el desarrollo de un sistema distribuido para la administración de calendarios y planificación de eventos, así como la metodología y motivaciones para su implementación.

Palabras Clave: Código abierto, estándares, comunidad, colaboración, interoperabilidad, sistema distribuido.

Abstract

The aim of this article is to propose the use of open source and open standards as one of the best current practices in the software and systems development area. It exposes the increasing global interest towards these tools, due to the high availability of Internet access that allows programmers globally distributed to form communities where the collaboration enables the production of high quality code, with fewer errors and less security vulnerabilities than proprietary software, and at a relative low cost.

It also covers the need to use open standards to ease the interoperability with other systems and applications.

In this paper the application of this model is focused in the development of a distributed system for the administration of calendars and event scheduling. The methodology and motivations for the implementation of this system are also presented.

Keywords: Open source, standards, community, collaboration, interoperability, distributed system.

1. Introducción

Durante sus primeros 40 años de existencia, la informática ha evolucionado a pasos agigantados, desde grandes máquinas a las cuales sólo grupos militares o académicos tenían acceso hasta los más avanzados y personales dispositivos de hoy en día. Ha migrado de la centralización de sistemas hacia los sistemas distribuidos globalmente, debido en gran medida al despliegue del Internet.

Así mismo la idea del software de código abierto (OSS, por sus siglas en inglés) no es nueva, sin embargo hoy en día el número de usuarios alrededor del mundo que contribuyen al OSS ha alcanzado niveles considerables; tal que un número creciente de compañías, organizaciones e incluso estados están usando o al menos analizando el uso del OSS y estándares abiertos para la comunicación e implementación de sus sistemas.

El OSS ha logrado producir algunos productos impresionantes; sus soluciones son empleadas diariamente sin siquiera notarlo: sistemas operativos como Linux, servidores de correo como Sendmail, servidores Web como Apache, bases de datos como MySQL y hasta sitios como Amazon y Google se han adherido al modelo de código abierto para brindar sus servicios.

La estandarización es otra área que ha alcanzado sorprendentes progresos, sobretodo en el área de comunicaciones, y que constituye un medio para reducir la complejidad y simplificar los procesos de trabajo. El siguiente reto consiste en habilitar los estándares a nivel de la información y de las interfaces de aplicaciones de software que intercambian esta información.

De acuerdo a lo expuesto, podemos enmarcar los objetivos y alcance del presente documento en los siguientes puntos.

- Explicar las características del código abierto que lo convierten en un modelo sostenible para el desarrollo de proyectos y el cual puede ajustarse a las necesidades individuales de cada programador.
- Establecer la necesidad de trabajar con estándares o especificaciones abiertas en el desarrollo de proyectos brindando así independencia tecnológica.
- Cubrir la metodología empleada en el desarrollo de un "Sistema de Calendarios Distribuido" cuya utilidad será mejorar el grado de comunicación y colaboración efectiva en la administración del tiempo siguiendo el estándar iCalendar.
- Contribuir al desarrollo del código abierto que permite ofrecer productos extensibles y de alta calidad.

2. Metodología

El software de código abierto (referido como OSS) [1] está basado en una idea relativamente sencilla: el núcleo del sistema es desarrollado localmente por un solo programador o un grupo de programadores. Un prototipo es liberado en Internet, cuyo código fuente puede ser leído, modificado o redistribuido por otros desarrolladores libremente [2]. Este mecanismo permite la evolución del sistema de una forma extremadamente rápida, más rápida que la secuencia normal de un modelo cerrado tradicional; en el cual solo un equipo de desarrollo de un limitado número de programadores tiene acceso y permiso de modificación sobre el código fuente.

El beneficio y ventaja principal del código abierto radica precisamente en su nombre: disponibilidad del código fuente. Desde el punto de vista de desarrollador esta constituye

una alternativa óptima para entender sistemas complejos y aplicar código aprendido en futuras soluciones.

El uso de código abierto en el desarrollo de proyectos evita reinventar la rueda. Es decir, se puede tomar partes de software ya escrito y que más se aproximen a nuestras necesidades para adaptarlas y convertirlas como plataforma de lanzamiento para nuevas aplicaciones, aún cuando posteriormente a medida del proceso de desarrollo la versión final difiera de la original siempre es más fácil partir de una solución parcial que de cero [3].

Entre los múltiples beneficios del modelo de desarrollo de OSS, se encuentra el hecho de que los errores son más evidentes si se exhiben simultáneamente a miles de desarrolladores; quienes al cooperar, descubrir o corregir errores permiten aumentar la eficiencia del sistema del cual ellos mismos son usuarios. Sin embargo, no sólo se obtienen beneficios relativos al código; sino también concernientes a la interacción con el usuario. Dado que muchos de ellos contribuyen aportando comentarios acerca de la usabilidad de los sistemas provistos, mejorando así el diseño de los mismos.

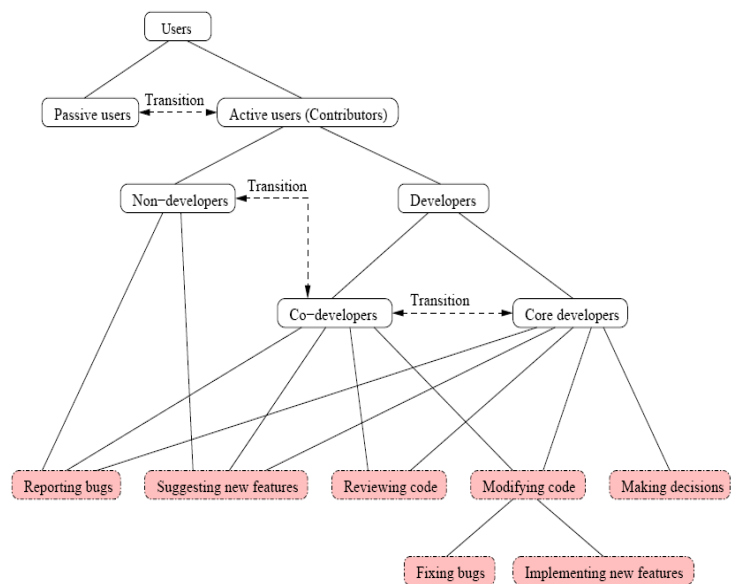


Figura 1. Clasificación de usuarios y desarrolladores de Open Source [4]

Los estándares abiertos por su parte, al consistir en especificaciones de alcance público, indican la pauta para que el sistema a desarrollar pueda ser compatible con diversos componentes de software existentes, evitando así problemas de comunicación o interoperabilidad por el uso de diferentes formatos, garantizando que el producto entregado sea sostenible en el largo plazo así como la conservación, pervivencia, integridad y reusabilidad de la información [5].

El formato iCalendar constituye el estándar abierto empleado en el desarrollo del sistema objeto de análisis en este documento.

Caso de Estudio: Calendario Distribuido

El presente proyecto intenta cubrir la planificación de eventos, tareas y entradas de un diario a través de una aplicación de escritorio que permita la administración local de los mismos; así como la compartición de esta información hacia un servidor remoto, a través del cual otros puedan conocer las actividades programadas de un individuo en particular, de un grupo de trabajo o de toda una institución, y que a su vez les permita suscribirse o solicitar citas de acuerdo a la disponibilidad de tiempo del sujeto en mención, las mismas que serán enviadas de manera remota por medio del correo electrónico, para su respectiva aceptación o rechazo por parte del dueño del calendario u organizador.

El formato de almacenamiento e intercambio de la información de planificación seguirá la norma RFC2445 o iCalendar [6]. El establecimiento de notificaciones se realizará por medio de alarmas sonoras de manera local en la aplicación de escritorio y de correos

electrónicos con posibilidad de ser enviados hacia un celular para la notificación remota de eventos y tareas programadas, así como para la solicitud de citas.

Estándares de Calendario: El Formato de Intercambio iCalendar

En 1996 se formó un grupo de trabajo dedicado al desarrollo de estándares de calendario y planificación en el Internet que permitieran la interoperabilidad de diferentes productos. El grupo se conoció bajo el nombre de IETF's Calendaring and Scheduling (CalSch) [7] [8], cuyos estándares se encuentran especificados en los siguientes RFCs:

- RFC 2445: Internet Calendaring and Scheduling Core Object Specification (iCalendar)
- RFC 2446: iCalendar Transport-Independent Interoperability Protocol (iTIP)
- RFC 2447: iCalendar Message-Based Interoperability Protocol (iMIP)
- Calendar Access Protocol (CAP)

El estándar 'iCalendar' especifica un formato común para el almacenamiento e intercambio de información tipo calendario a través de innumerables medios de transporte, incluyendo pero no limitado a SMTP, HTTP, FTP. El formato consiste en una secuencia de propiedades y uno o más componentes de calendario. Las propiedades son atributos que se aplican a todo el calendario. Los componentes son colecciones de propiedades que expresan una semántica particular en el lenguaje del calendario. Por ejemplo: el componente de calendario puede especificar un evento, una tarea, una entrada del diario, información de zona horaria, información de tiempo disponible/ocupado o una alarma.

Requerimientos del Sistema

El administrador de Calendario requiere de los siguientes elementos para brindar su funcionalidad:

- Cliente de Calendario: Mozilla Sunbird
- Servidor de Calendarios y Front-End Web
- Base de Datos de información del calendario y estructura del servidor
- Soporte de establecimiento de citas y envío de recordatorios
- Soporte de solicitud de citas de manera remota
- Suscripción a otros calendarios
- Información recuperable en formato iCalendar

❖ Cliente de Calendario: Mozilla Sunbird

Mozilla Sunbird [9] es una aplicación independiente basada en la extensión de calendario de Mozilla y que cumple funciones de agenda, lista de tareas y calendario con alarmas. Dispone de una interfaz de diseño sencillo, con varias posibilidades de visualización y funciones que permiten gestionar la vida diaria: programación de tareas, citas, aniversarios y otros eventos importantes, herramienta de alarmas, etc.

Tomando como base la idea inicial de evitar reinventar la rueda, el cliente del sistema consiste en colaborar con la comunidad de código abierto de Sunbird en el cumplimiento de los requerimientos planteados de la aplicación a fin de convertirlo en una alternativa viable de los productos que soportan la norma iCalendar existentes en la red.

Sunbird se encuentra desarrollada primordialmente en dos lenguajes: XUL y Javascript.

XUL es un lenguaje basado en XML utilizado para describir y crear interfaces de usuario, que ha sido diseñado para brindar la portabilidad de las mismas, por lo que permite desarrollar aplicaciones multi-plataforma sofisticadas o complejas sin necesidad de

herramientas especiales. Así una interfaz puede ser implementada y modificada fácil y rápidamente.

El lenguaje Javascript en cambio es empleado para manejar el comportamiento de la interfaz de usuario.

Sunbird almacena la información de calendario por medio de SQLite. SQLite constituye una librería escrita en lenguaje C que implementa un manejador de base de datos SQL embebido, de forma tal que permite tener acceso a una base de datos SQL sin tener que ejecutar un programa RDBMS separado.

Sunbird tiene entre otras las siguientes características: trabaja con archivos basados en el estándar iCalendar; permite importar, exportar o suscribirse a archivos iCal; permite publicar calendarios hacia un servidor FTP o WebDAV.

❖ Servidor de Calendarios: Front End Web y Base de Datos MySQL

El servidor de calendarios permite la distribución y/o promoción de la información de planificación en línea. Para ello, requiere de un front-end web el cual se halla implementado en código PHP, el mismo que permite la visualización de los calendarios publicados en el servidor de manera individual o colectiva, así como la solicitud de citas o reuniones en un horario determinado de un calendario.

El servidor trabaja paralelamente con una base datos MySQL, la cual contiene la estructura necesaria para almacenar tanto la información relativa al calendario como las entidades que permiten brindar características adicionales al sistema tales como listas de contactos, compartición, publicación de eventos entre otras.

El sitio del sistema permite la visualización de la planificación personal por día, semana o año. Ofrece la posibilidad de descargar la información de un calendario completo como un archivo de extensión "ics", es decir que cumpla la norma RFC2445, el cual puede ser interoperar con otras aplicaciones de escritorio o web que se apeguen al estándar.



Figura 2. Sección de Administración del Sistema

Características del Servidor:

- Ilimitado número de calendarios: Cada usuario registrado en el sistema puede poseer varios calendarios para agrupar a sus eventos, tareas o diarios.
- Ilimitado número de categorías: Las categorías pueden existir de manera global o relativa a cada usuario.
- Interfaz intuitiva y fácil de usar: La navegación del sistema le permite hallar su información de planificación fácilmente y de forma efectiva. Los eventos en el calendario corresponden al color del calendario en la lista

- Múltiples Vistas: La información de calendarios puede visualizarse de forma diaria, semanal, mensual y anual.

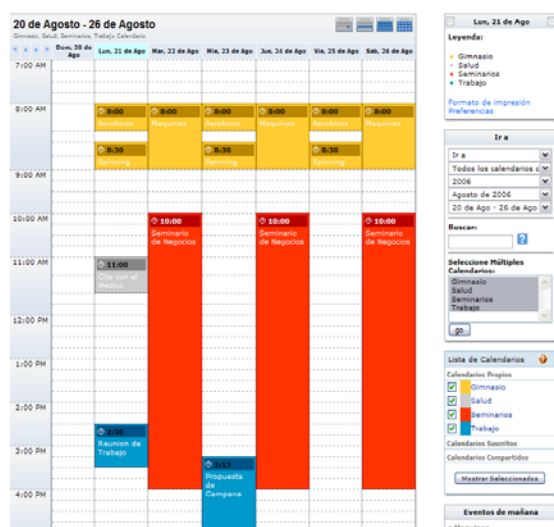


Figura 3. Presentación de Calendarios Combinados en Vista Semanal

Un calendario puede ser exportado en formato iCalendar con todos los eventos, calendarios, diarios descritos en el mismo.

- Suscripciones de Calendarios: Los usuarios pueden suscribirse a los calendarios seleccionados y recibir recordatorios o notificaciones por correo de los eventos de los mismos.
- Impresión de Calendarios: Las vistas pueden transformarse en listas de eventos para su impresión.
- Entradas recurrentes: Los usuarios pueden indicar la recurrencia de las entradas indicadas en sus calendarios.
- Panel de Administración: Accedida por el administrador, permite el manejo y configuración de todas las preferencias y elementos existentes en el sistema.
- Editor WYSIWYG¹ para journals: Los journals constituyen en entradas visibles solo para el que las generó y permiten la aplicación de formato a su contenido.
- Participantes de Eventos y Tareas: Los usuarios pueden elegir de entre sus contactos a aquellos que serán participantes de sus entradas.
- Múltiples Administradores: El sistema puede tener varios administradores, indicando el tipo de usuario creado.
- Personalización de Preferencias: Configure las preferencias del sistema.
- Administración de Categorías: Habilidad de Administrar categorías registradas por cada usuario.
- Compartición de calendarios: Se puede compartir la administración de un calendario con otros usuarios del sistema con diferentes niveles de permisos.

3. Resultados y Discusión

Desde 1998, con la aparición de estándares abiertos, los vendedores de aplicaciones de calendario han encaminado sus esfuerzos en proveer productos que se basen en estándares. Sin embargo, iCalendar no ha logrado el éxito esperado [10].

¹ WYSIWYG: Acrónimo de What You See Is What You Get (en inglés, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.

La habilidad para intercambiar calendarios de un producto a otro, depende ampliamente en la medida en la cual el producto pertinente soporte el formato iCalendar. La mayoría de ellos soportan un subconjunto de la especificación y el subconjunto soportado por uno de ellos generalmente diferente del subconjunto soportado por otro.

Entre las características comunes de los productos actuales tenemos que la mayoría soporta el intercambio simple de eventos y alarmas, mientras una menor cantidad soporta los objetos Todo, Journal.

Asimismo, las aplicaciones reaccionan de forma distinta frente a información inmanejable para ellas; pudiendo degradarse naturalmente aceptando la información entendible, o producir una notificación apropiada acerca de la información que no pudo ser importada exitosamente.

El proyecto objeto de este estudio muestra que todos los usuarios se beneficiarían de un sistema que abarcara a toda la facultad y que les permitiera ver y administrar sus eventos públicos, privados y grupales en un solo lugar. Así por ejemplo, la facultad podría requerir de su propio 'diario departamental' para registrar eventos que se aplican sólo a ellos, como fiestas, fechas límite o reuniones. Los departamentos o unidades podrían necesitar agregar o ver los eventos disponibles. Finalmente, cada individuo querría un diario personal para anotar sus eventos sociales, almuerzos y demás.

Esta actividad de planificación toma tiempo al realizarse de manera independiente, y más aún sosteniendo un registro manual de los eventos o tareas.

La solución plantea el registro de eventos y/o cronograma de actividades y habilita el intercambio de los mismos con otros usuarios del sistema o su publicación en la red, a fin de que los eventos públicos pudieran ser observados por otros, quienes a su vez tendrían la oportunidad de solicitar citas de acuerdo a su disponibilidad de tiempo.

Debe considerarse además que el problema relacionado con la planificación pudiera haber sido resuelto parcialmente por algún departamento, grupo o individuo, por lo cual el nuevo sistema de planificación debe ser modular y compatible con los sistemas de calendario existentes.

4. Conclusiones

En el presente trabajo se ha tratado de aclarar lo que constituye el código libre, su metodología de desarrollo basada en comunidades y en la motivación personal de los desarrolladores involucrados en este tipo de proyectos para producir software que satisfaga sus necesidades personales y que a la vez sea altamente productivo.

El software de código abierto otorga a los usuarios varias posibles soluciones de satisfacer mejor sus requerimientos de computación, beneficio que ha captado hoy por hoy la atención de empresas de software a nivel mundial.

Finalmente el open source permite construir aplicaciones en las cuales se pueda añadir extensiones fácilmente y sin pérdidas de estabilidad y calidad, provee independencia de vendedores de software, menores costos de implementación y facilidad de acoplamiento con otros sistemas.

5. Trabajo Futuro

El componente Web del sistema de Administración de Calendarios distribuidos puede ser integrado fácilmente a aplicaciones existentes tanto de la facultad como de la Universidad entera. Así, puede enlazarse con el Sistema de Comunicación Alumno-Profesor METIS donde su aplicación principalmente se encaminaría hacia un intercambio

de doble vía de datos de alumnos pertenecientes a un curso dado para mejor planificación de eventos grupales.

En un contexto mayor, el sistema puede integrarse al actual SIDWEB de forma tal que los calendarios de distintas unidades, facultades o institutos pudiesen consultarse en línea y la programación se realizara en la totalidad de la universidad. Dicho esto, los usuarios deberían ser miembros de la ESPOL para lo cual el sistema debería vincularse a la base de usuarios existentes.

Como características adicionales puede incorporarse la capacidad de publicar blogs y aceptar comentarios de otros usuarios del sistema.

6. Bibliografía

- [1] OPEN SOURCE SITE. *Sitio Oficial del Open Source* [en línea]. Disponible en web: <<http://www.opensource.org/>>
- [2] STAMELOS, J; ANGELIS L; OIKONOMOU, A; BLERIS, G. *Code quality analysis in open source software development*. Disponible en web: <<http://turingmachine.org/opensource/papers/stamelos2002.pdf>>
- [3] RAYMOND, Eric. *The Cathedral and the Bazaar* [en línea]. Septiembre 11, 2000. Disponible en web: <<http://catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html>>
- [4] GACEK C; ARIEF B. *The Many Meanings of Open Source*. IEEE Software, Enero/Febrero, 2004, vol. 21, no. 1, p. 34-40. Disponible en línea: <<http://portal.acm.org/citation.cfm?coll=GUIDE&dl=GUIDE&id=968793>>
- [5] ESTANDARES ABIERTOS. *Manifiesto (Beta 2)* [en línea]. Marzo 9, 2006. Disponible en web: <http://www.csc.com/features/2004/uploads/LEF_OPENSOURCE.pdf>
- [6] DAWSON, F; STENERSON, D. *Internet Calendaring and Scheduling Core Object Specification (iCalendar), RFC 2445* [en línea]. Nov. 1998. Disponible en: <<http://www.ietf.org/rfc/rfc2445.txt>>
- [7] DAWSON, Frank. *Emerging Calendaring and Scheduling Standards* [en línea]. IEEE Computer, Diciembre 1997, vol. 30, n° 12, p. 126-128, Transcripción y conversión al formato HTML de la red.: Charles Severance, Michigan State University. Disponible en versiones PDF y HTM en: <<http://csdl2.computer.org/dl/mags/co/1997/12/rz126.htm>>
- [8] NETSCAPE COMMUNICATIONS. *More than 20 Companies Join Netscape to Help Define Open Standard for Internet Calendaring and Scheduling* [en línea]. 24 Julio 1996. Disponible en: <<http://wp.netscape.com/newsref/pr/newsrelease194.html>>
- [9] MOZILLA SUNBIRD PROJECT. *Sitio Oficial Mozilla Sunbird* [en línea]. Disponible en web: <<http://www.mozilla.org/projects/calendar/sunbird/>>
- [10] TECHTALKS EVENT. *Calendaring: What We Know, What We Don't Know* [en línea]. Mayo 11, 2000, actualización Dic. 5, 2005. Invitados expertos: Paul B. Hill y Bob Mahoney. Co-anfitriones: Howard Strauss, Judith Boettcher. Disponible en web: <<http://www.campus-technology.com/techtalks/events/000511calendaring.asp>>
Formato de audio en: <<http://www.campus-technology.com/techtalks/Audio/2000/cren051100.mp3>>