Distribución y administración de contenidos de sitios Web utilizando técnicas de replicación y sincronización

Paola Bonilla Bermeo [1], Gerardo Garnica Benítez [2], Fabricio Echeverría [3]

1 Ingeniera en Computación Sistemas de Información 2005
2 Ingeniero en Computación Sistemas de Información 2005
3 Director de Tesis. Ingeniero en Computación, Escuela Superior Politécnica del Litoral, 1999, Profesor de ESPOL desde 2000.

Resumen

En la actualidad, la forma de trabajar está cambiando veloz y radicalmente y la información es uno de los activos más importantes que poseen las organizaciones, para lo cual utilizan soluciones que cubren dichas necesidades. De esta manera, el reto es presentar estas tecnologías y aplicaciones como un todo unificado y accesible a través del Internet, para que esté disponible en todo momento a diferentes tipos de usuarios. Igualmente, es importante desarrollar aplicaciones que manejen información dentro de una base de datos distribuida, en donde el contenido no esté centralizado en un solo sitio y pueda ser ingresado y modificado desde el lugar donde se produce la transacción y luego, a través del proceso de sincronización, actualizar los otros repositorios de datos de una organización. De esta manera, se presenta una solución que combina el uso de un sistema administrador de contenidos de un negocio de arriendo de inmuebles y que posee varios lugares en donde se almacenan los datos, con un servicio de sincronización para distribuir las modificaciones realizadas durante una transacción de un lugar a otro.

Abstract

At the present time, the way of working is changing quickly and radically and the information is one of the most important assets of the organizations; for that reason, they use some solutions that cover these necessities. So, the challenge is to display these technologies and applications like an unified and accessible whole thing, through the Internet that it is available at any moment to different kinds of users. Also, it is important to develop some applications that handle information within a distributed database where the content is not centralized in a single site and it can be entered and be modified from the place where the transaction takes place and then, through the process of data synchronization, to update the other data store of an organization. In the same way, a solution appears that combines the use of a content management system of leasing business, which it has several places in where the data are stored, with a synchronization service to distribute the modifications made during a transaction from a place to another one.

1. Introducción

Este trabajo tiene como objetivo el desarrollo de una solución que administre los contenidos de un sitio Web, y que con el uso de técnicas de sincronización, facilite la transmisión y distribución de la información que se encuentra en diferentes repositorios de datos que poseen la misma estructura de almacenamiento para los mismos. De esta forma se permite a los usuarios acceder a los contenidos y datos del sitio Web desde cualquier lugar donde opere el negocio y paralelamente, éstos, dependiendo de su nivel de acceso, pueden manipular la información del mismo, con la seguridad de que ésta será replicada a todos los puntos de negocio conectados de forma automática.

La solución desarrollada, consiste en un sistema administrador de contenidos de un negocio de alquiler de inmuebles; este negocio mantiene puntos de administración en diferentes lugares, los cuales

poseen las mismas estructuras de almacenamiento de datos, y cuyas transacciones se distribuyen a los otros sitios a través de un servicio de sincronización. Este servicio replica la información de cada uno de los registros de las tablas que hayan sido objeto de alguna transacción, luego de la última vez que se ejecutó el servicio. De esta forma se provee una solución que optimiza los recursos del sistema como tiempo y dinero, mediante rapidez y eficacia en la manipulación de la información.

2. Fundamentos teóricos

2.1. Administradores de contenidos

Los sistemas administradores de contenidos (Content Management System, CMS) son aplicaciones hospedadas en un servidor Web que permiten la creación, administración, distribución y publicación de información, lo que posibilita administrar el contenido de un sitio con facilidad y conveniencia. Los CMS emplean una interfaz que no requiere de conocimientos específicos de desarrollo Web para su manejo y en general se basan en una plantilla diseñada de antemano que actúa como una plataforma para cada página del sitio Web.

Figura 1. Flujo de información a través de un CMS.

La utilización de esta herramienta, ahorra personal especializado y tiempo en el ingreso de nuevo material en un sitio Web y permite al usuario concentrarse en el contenido y no en la parte técnica de la elaboración de un sitio, la cual es realizada por programadores. Además, una amplia gama de contenidos pueden ser publicados a través de un CMS, como páginas simples o complejas con información dinámica que proviene de bases de datos, manuales en línea, documentos generales de negocios, etc.

Las compañías que requieren el uso de un CMS por lo general son organizaciones en donde la publicación en Web está distribuida en muchos sitios y en la cual comunicar el contenido entre los mismos consume mucho tiempo, o que poseen un sitio Web en donde se requiere de frecuentes actualizaciones de contenido y estructura.

2.2. Replicación y sincronización de datos

La replicación de datos consiste en la transmisión de los mismos entre dos o más instancias de servidores, permitiendo transferir la información a más de un sitio. Su principal ventaja radica en aumentar la disponibilidad de los datos y mejorar el rendimiento de las consultas globales. Sin embargo, su desventaja se encuentra en que debido a que afecta al log de la base de datos, torna el proceso de implementación más complejo.

La sincronización de datos es un mecanismo similar que, a diferencia de la replicación, éste afecta directamente a la capa de datos, con lo cual se reduce la complejidad en el desarrollo de la aplicación, pero crea una responsabilidad adicional en el manejo de la seguridad de los datos.

Así pues, para la distribución oportuna de los datos se ha escogido la técnica de sincronización, puesto que en la actualidad aporta un enfoque innovador al complejo y tedioso proceso de distribución de la información entre un servidor central y las bases de datos remotas.

3. Análisis del sistema

3.1. Requerimientos

Para efectos de demostración se ha desarrollado un sistema administrador de bienes raíces que, a través de una interfaz Web, permite llevar un control de arriendos de inmuebles, tales como departamentos, villas, oficinas y locales, en el cual se tiene un registro de todos los propietarios y arrendatarios de los que están siendo alquilados. De la misma manera, esta aplicación permite realizar un seguimiento de diferentes procesos del negocio, entre ellos constan los pagos de los arriendos mensuales por parte de los arrendatarios, y una serie de consultas con el fin de simplificar cada uno de dichos procesos.

Además, existe el módulo de administración de contenidos para facilitar las tareas de los usuarios en la manipulación del sitio. Para este efecto se han definido 3 tipos de usuarios:

- Administradores, quienes tienen la capacidad de administrar todo el contenido del sitio, y a su vez manejan los permisos de los usuarios y de los grupos, así como definen las configuraciones de seguridad.

- Agentes operadores de bienes raíces, que son los encargados de la publicación de inmuebles que están disponibles para ser alquilados, así como del ingreso, modificación y eliminación de arrendadores, arrendatarios, garantes e inmuebles. Adicionalmente, son encargados del proceso de generación de contratos y del pago de arriendos.

- Usuarios públicos, quienes tienen acceso a la información del sitio Web, mas no a la manipulación del contenido.

Finalmente, como se desea lograr consolidación de datos en diferentes sitios y justo a tiempo, el sistema provee un módulo de sincronización de datos, que es manejado por los administradores del sistema.

3.2. Tecnologías

Como uno de los objetivos de este trabajo es facilitar la transmisión y distribución de los contenidos independientemente de la plataforma en la que se han desarrollado, el sistema operativo bajo el cual funciona este proyecto puede ser indistintamente Windows o Linux.

En la misma forma, se necesita un servidor Web que pueda operar indiferentemente en cualquier plataforma y que posea un buen rendimiento. Se ha escogido el servidor Web Apache debido a que éste es un servidor que puede ser utilizado tanto en Windows como en Linux, a diferencia de Internet Information Server (IIS) que sólo opera bajo Windows. Además, Apache no necesita licenciamiento, posee buen rendimiento, alta estabilidad y solidez, y en ambiente Windows el proceso de instalación y configuración es proporcionado por un wizard.

Adicionalmente, se ha optado por el uso de PHP (lenguaje de servidor para desarrollo Web) debido a su flexibilidad, facilidad de uso y mantenimiento, e independencia de plataforma, lo cual permite desarrollar una aplicación que sincroniza datos independientemente de la plataforma que se esté utilizando. Además permite la implementación de clases como cualquier lenguaje de programación orientado a objetos, lo que permite encapsular la lógica de negocios de una aplicación y separarla del contenido.

Por otro lado, para el desarrollo del servicio de transferencia de datos entre diferentes nodos, se ha considerado crear un servicio de sincronización entre las diferentes bases de datos del sistema distribuido debido a las ventajas indicadas anteriormente. Para tal fin se ha optado por desarrollar un mecanismo de sincronización de datos distribuido en la misma herramienta que el sitio Web (PHP) que se comunique a través de una red (que puede ser LAN o WAN), en donde cada uno de los puntos repositorios de datos del modelo distribuido mantenga su información y envíe a los otros repositorios las transacciones que se van produciendo tal como se muestra en la Figura 2.

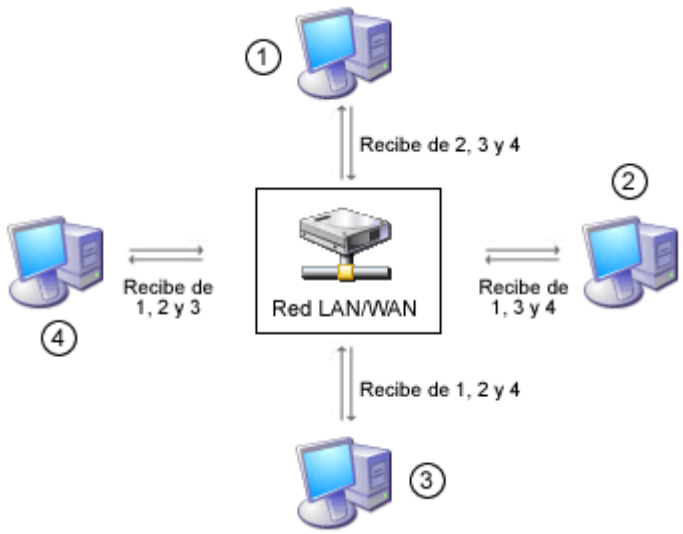

Figura 2. Esquema de sincronización distribuida:
Los nodos envían y reciben datos entre ellos directamente

4. Diseño del sistema

4.1. Arquitectura del sistema

En primera instancia se describe el servidor de los nodos locales que luego se conecta a otros nodos similares para sincronizar sus datos (Figura 3). Los servidores nodos contienen las aplicaciones necesarias que permiten atender los requerimientos de las páginas dinámicas a las que van a acceder los usuarios.

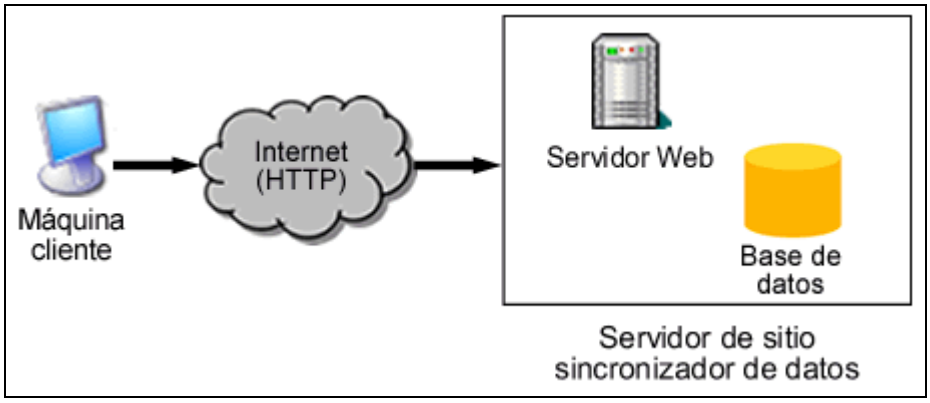

Figura 3. Arquitectura de un nodo sincronizador de datos

De igual manera estos servidores poseen los datos con los cuales interactúan los usuarios de una red específica. De esta forma, cuando un usuario cliente desea acceder al sistema a través de un browser, hace un requerimiento al servidor Web, el cual transmite la página de inicio del sitio Web, la cual ya contiene información proporcionada por la base de datos del sitio. Si el usuario desea ingresar al sistema con su nombre de usuario y contraseña, envía el pedido al servidor Web el cual verifica los datos y, si éstos son correctos, establece una sesión entre el cliente y el servidor Web.

A partir de este momento el usuario ejecuta las transacciones que requiera, envía el requerimiento al servidor Web, quien lo procesa conectándose a la base de datos y devuelve la respuesta al cliente. Cuando el usuario deja de utilizar el sistema se cierra la sesión establecida entre el cliente y el servidor.

De la misma forma, el modelo bajo el cual se ha considerado crear el servicio de sincronización en este sistema distribuido, toma en consideración que cada nodo compuesto por el conjunto de máquinas clientes, servidores y base de datos, debe enlazarse con los otros nodos, para lo cual, debe utilizar el protocolo TCP/IP para la comunicación y el envío de datos entre estos componentes, tal como se muestra en la figura 4.

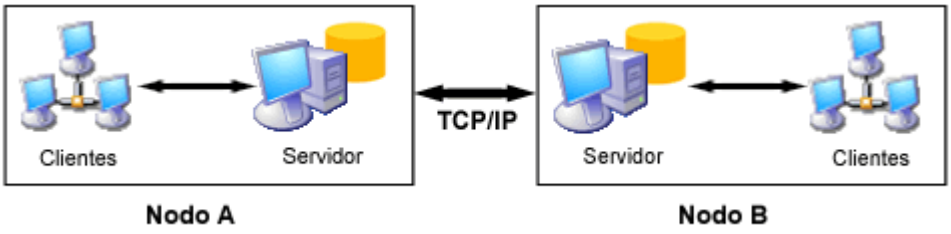

Figura 4. Arquitectura del modelo de sincronización

4.2. Diseño de la base de datos distribuida

Así pues, se requiere el desarrollo de un modelo de datos distribuido, en el que se ha considerado dividir el diseño de la base de datos en 2 modelos:

- Modelo de datos de los contenidos.
- Modelo de datos del negocio.

Esta división se ha hecho con el fin de que el modelamiento del sistema permita la independencia en el manejo de la presentación de la información, es decir, la administración de contenidos, con respecto

a los datos propios de una organización, lo que permite que el diseño de base de datos propuesto se adapte a cualquier modelo de negocios manteniendo la estructura de los contenidos. Una de las ventajas de este enfoque es que el diseño del modelo de contenidos se adapta a cualquier tipo de tecnología, desde páginas simples hasta información altamente dinámica que se origina de una base de datos que posee un elevado número de transacciones en línea.

En el modelo de datos de los contenidos se almacenan todos los elementos que permiten la administración de la información que da soporte a la presentación de los datos del negocio. Es en este modelo donde se manejan las opciones que posee el sistema con sus elementos de manipulación de la información, así como los usuarios y permisos que restringen el acceso a las opciones del sistema.

El modelo de datos del negocio contiene toda la información propia de las transacciones que se ejecutan dentro de una organización y que tienen importancia en la gestión del negocio, así como, en la arquitectura Cliente – Servidor de 3 capas componen la capa de lógica de negocios.

Un aspecto importante para decidir el motor de bases de datos que se utilizó en este proyecto, es que éste debe estar disponible para prácticamente cualquier plataforma. Por tanto, se ha escogido como herramienta para el almacenamiento de los datos PostgreSQL, el cual es un sistema administrador de bases de datos relacional (RDBMS), open–source, que soporta diferentes funcionalidades, tales como querys complejos, disparadores (triggers), vistas, integridad relacional y seguridad de las transacciones. Una ventaja adicional que proporciona el uso de PostgreSQL es que puede ser extendido de muchas maneras, por ejemplo, añadiendo nuevos tipos de datos, funciones, operadores, funciones agregadas, o lenguajes procedurales.

Adicionalmente, debido a su licencia abierta, PostgreSQL puede ser utilizado, modificado y distribuido libremente, y toda la documentación esta disponible gratuitamente. Existen versiones de PostgreSQL para Linux, Windows, Sun Solaris, IBM AIX y FreeBSD, lo que asegura la facilidad de usar los diseños planteados bajo cualquier plataforma. Por estos motivos se ha considerado el uso de PostgreSQL para el almacenamiento de las tablas y los datos de los modelos de datos de contenidos y negocio del sistema desarrollado.

4.3. Consideraciones de diseño de la base de datos distribuida

Es necesario diseñar las tablas que van a sincronizar de tal modo que al momento de que los datos tengan que pasar de un repositorio a otro no surjan conflictos de replicación. Esto podría ocurrir, por ejemplo, cuando en uno de los repositorios se ingrese un registro de arrendador con identificación ‘0951264161’, y antes de que los datos sincronicen, en otro repositorio también se produzca el mismo ingreso de un registro de arrendador. Esto ocasionaría un fallo en la transferencia de datos cuando se produzca la sincronización porque no se podría copiar el nuevo registro de la base origen a la base destino. Para evitar este conflicto se ha adoptado que las tablas que vayan a sincronizar posean una clave primaria compuesta, la cual estaría formada por el identificador propio de un registro de la tabla y un código de la base de datos de origen.

Este código estará compuesto de 3 caracteres con el siguiente formato: ‘TE#’, donde # será un dígito del 1 al 9, diferente para cada repositorio de datos, lo que permitirá indicar e identificar en qué base fue creado el registro. Además, las tablas que necesiten un identificador que en un modelo de datos no distribuido debiera ser secuencial, en este modelo tendrán que tener un identificador de tipo entero, es decir, que no se utilizará el tipo de dato secuencial.

Adicionalmente, se han agregado 2 campos a las tablas del modelo distribuido:

Operación: Indica si se ha registrado alguna operación en el registro, lo cual permitirá al sincronizador determinar si este registro tiene que ser enviado a los otros repositorios de datos. Si el registro ha sido creado y aún no ha sido replicado, su campo operación será marcado con ‘I’ (de “Insert”). Si el registro ha sido modificado y aún no ha sido replicado, su campo operación será marcado con ‘U’ (de

"Update"). Una vez que el sincronizador haya enviado los datos y actualizado las tablas de los otros repositorios de datos, el valor del campo operación de los registros marcados será cambiado a "Nulo" (null), lo cual indicará que el registro ya ha sido sincronizado.

Fecha de eliminación: Indica la fecha en que un registro ha sido eliminado de la base de datos. Es preciso anotar que, al tratarse de un modelo de datos distribuido, no se puede aplicar el borrado físico de los registros de la base de datos, por lo que es necesario aplicar la eliminación lógica de los mismos. La eliminación lógica de un registro significa que cuando un usuario que utiliza el sistema elimina un registro, por ejemplo de arrendatario, éste registro no desaparece físicamente de la base de datos sino que es marcado como "borrado"; para esta transacción se actualiza el valor del campo 'Estado' de cada tabla como "Inactivo" o "Eliminado", se marca el campo 'Operación' con 'U' (de "Update") y se registra en el campo 'Fecha de Eliminación' la fecha en que se ha producido la transacción. El hecho de establecer el campo 'Fecha de Eliminación' permitirá que en un momento determinado una aplicación adicional del sistema elimine físicamente de la base de datos los registros cuya fecha de eliminación exceda cierto tiempo que ya hace inútil su conservación dentro de las tablas de la base.

Adicionalmente, se ha creado una tabla que va a contener la información de las imágenes relacionadas a los inmuebles registrados en el sistema, debido a que cuando un usuario adjunta una imagen en un registro de inmueble, esta tiene que ser replicada a los otros repositorios de datos, lo cual se debe realizar a través del servicio de sincronización para que la imagen sea transferida a los otros servidores del modelo distribuido.

4.4. Diseño del sincronizador de datos

El sincronizador de datos deberá tener en cuenta dos aspectos para su funcionamiento: se deberá ejecutar constantemente como un proceso en segundo plano dentro del sistema operativo donde se encuentra el servidor de la base de datos, y deberá establecer un mecanismo de comunicación con los otros repositorios de datos con los que va a sincronizar cada uno de los registros de las tablas que tengan marcado el campo 'Operación'.

Las comunicaciones entre los puntos de origen y destino, el cual puede ser más de uno, están establecidas dentro de un archivo de configuración. Este archivo posee, en la primera línea, los parámetros de acceso a la base de datos desde la cual van a sincronizarse los registros, y en las líneas siguientes, la configuración de los parámetros de acceso a las bases de datos hacia las cuales van a llegar los registros sincronizados desde la base de origen. El programa que ejecuta el servicio de sincronización accede a este archivo y establece las conexiones respectivas desde la base de origen a las bases de destino.

Una vez establecidas las conexiones a cada punto del sistema distribuido se establecen llamadas a los procedimientos que transfieren los datos por sincronizar de la base de origen a las bases de destino. Se ha preferido establecer un procedimiento por cada tabla que vaya a sincronizar sus datos para permitir la escalabilidad del programa cuando se requiera añadir nuevas tablas que estén habilitadas para sincronizar. Estos procedimientos hacen una consulta a la tabla respectiva de aquellos registros que tengan marcado el campo 'Operación', para reconocer si existe algún registro que haya sido recientemente ingresado o actualizado.

Si se obtienen resultados de la consulta a la tabla, entonces se realiza el ingreso o la actualización de cada registro encontrado hacia las bases de destino. Luego de finalizado el proceso en las bases destino, se hace una actualización del campo 'Operación' en los registros sincronizados en la base de origen y se desmarca dicho campo, lo cual indica que el registro ya ha sido sincronizado.

5. Conclusiones

- Como resultado de haber desarrollado este proyecto se tiene un sistema administrador de bienes raíces que, principalmente, combina una aplicación para administrar los contenidos de su sitio Web, con un servicio que sincroniza todas las transacciones generadas en cualquiera de los servidores que forman parte del negocio. Se ha enfocado el sistema hacia negocios en los cuales se maneje flujos de información muy variable y en donde se requiera optimizar recursos como tiempo y costos.

- Con el fin de cumplir con el objetivo de asegurar la independencia de plataforma que debe poseer este proyecto, la solución se implementó en herramientas que aseguraron la portabilidad y escalabilidad del mismo.

- El código de la aplicación ha sido diseñado y estructurado de tal manera que facilita la implementación de nuevos módulos al sistema de forma rápida y sencilla.

- La base de datos fue desarrollada de tal forma que se logró mantener la consistencia de datos y la integridad relacional de las tablas partícipes de la sincronización de los registros.

- El dividir el proyecto en dos modelos de datos, uno de negocios y otro de contenidos, permite separar los aspectos relacionados a cualquier negocio en sí, en este caso el de bienes raíces, de todo lo relacionado con la presentación de contenidos que pueden ser aplicados a cualquier tipo de negocios, y de esta forma se combinan dos poderosas herramientas en una misma aplicación.

- La solución encontrada para la resolución de conflictos de clave primaria, utilizando claves compuestas, mantuvo la integridad de los datos y eliminó los posibles problemas al momento de sincronizar la información.

- El tratamiento de imágenes utilizando una tabla específica para este fin, aseguró la reducción de los tiempos de consulta y facilitó la transmisión de datos al momento de sincronizar los mismos.

- Los niveles establecidos en la administración de contenidos, aseguraron una división apropiada de las tareas para cada usuario y a la vez proporcionaron la seguridad necesaria para evitar la manipulación de información restringida por parte de usuarios no autorizados.

- El crear un archivo para la configuración de las bases de datos dentro del modelo distribuido, que no forme parte de la base local, ni de la implementación del sitio Web, permitió la independencia de las operaciones del servicio de sincronización y, adicionalmente, brindó flexibilidad para la creación de nuevos puntos de destino.

- La implementación del servicio de sincronización bajo la misma plataforma en la cual fue desarrollado el sitio Web, permitió el uso del conocimiento adquirido en la realización de este proyecto y adicionalmente, aseguró la independencia de plataformas. Por otra parte, el sincronizador brinda la flexibilidad a un usuario administrador de configurar este servicio de acuerdo a las necesidades del negocio.

- El sistema ha sido desarrollado de tal forma que le permite a los nuevos usuarios la posibilidad de extenderlo conforme exija el negocio. Por lo tanto, es recomendable el desarrollo de proyectos futuros que tomen como referencia este trabajo.

- Es recomendable generar un módulo que recorra todos los registros que se encuentren inactivos e inutilizados durante un determinado período, para eliminarlos físicamente de la base de

datos, puesto que en este proyecto, se utilizó solo el concepto de eliminación lógica de registros.

6. Referencias

1. P. Bonilla, G. Garnica. Distribución y administración de contenidos de sitios Web utilizando técnicas de replicación y sincronización (Tesis, Facultad de Ingeniería en Electricidad y Computación. Escuela Superior Politécnica del Litoral, 2005).

2. M. Buretta. Data Replication: Tools and techniques for managing distributed information (Estados Unidos, John Wiley & Sons, 1997).

3. A. Dix, J. Finlay, G. Abowd, R. Beale. Human Computer Interaction (Gran Bretaña, Prentice Hall, 1998).

4. ERPToday.com. 2004, Content Management Tutorial, http://erptoday.com/CMS/Content-Management-Tutorial.aspx.

5. Y. T. Lau. The Art of Objects: Object–oriented design and architecture (Estados Unidos, Addison-Wesley, 2000).

6. Mtbase – Sybase de Colombia. 2004, Replicación de Datos y Warm Standby con Sybase Replication Server, http://www.mtbase.com. Co/pdf/ha/RS-Stdby.pdf.

7. D. Castro Morell, R. Pérez Váquez. Replicación de datos en SQL Server, http://www. monografias.com/trabajos15/ replicacion-datos/replicacion-datos.shtml.

8. Urudata Monthly Report. Replicación de datos a nivel corporativo, http://www.cp.com.uy/92/92-replicacion-datos.htm.

9. C. Sánchez. Técnicas de sincronización de datos, http://www.aui.es/biblio/bolet/bole024/art_mov_intesys.htm.