

Un Sistema de Vigilancia en Salud Pública para alertas tempranas

Diego M. López

Departamento de Telemática, Universidad del Cauca – Colombia.
dmlopez@unicauca.edu.co

Fecha de recepción: 10-1-2007

Fecha de selección: 8-10-2007

Fecha de aceptación: 30-8-2007

ABSTRACT

The paper describes a generic software architecture for Public Health Surveillance information Systems and the implementation of a reporting system for infectious diseases in rural areas, implementing such architecture. The architecture is a component-based, interoperable, flexible and open approach and the system is implemented using open source technologies. The software application has been piloted in the Department of Cauca in Colombia, in the framework of a telemedicine Project funded by the European Union Commission.

KEY WORDS

Software architecture, component-based development, XML, Public Health Surveillance.

RESUMEN

En este artículo se describe en detalle una arquitectura genérica para Sistemas de Vigilancia en Salud Pública y la implementación de un sistema de vigilancia de alertas tempranas en zonas rurales de países en desarrollo que utiliza la arquitectura propuesta. El sistema cumple con los requisitos de reusabilidad, desarrollo basado en componentes, flexibilidad, interoperabilidad y utilización de tecnologías abiertas. Esta herramienta fue piloteada en el departamento del Cauca, en Colombia, en el marco de un proyecto de investigación y desarrollo financiado por la Comisión Europea.

PALABRAS CLAVE

Arquitectura software, desarrollo basado en componentes, XML, Vigilancia en Salud Pública.

Clasificación Colciencias: Tipo 1

I. INTRODUCCIÓN

En Colombia el Sistema de Vigilancia en Salud Pública (Sivigila) es reglamentado por el Ministerio de la Protección Social y ejecutado a través los Planes de Atención Básica —PAB. En la Circular externa número 00018 de 2004 del Ministerio de la Protección Social [1], donde se establecen los lineamientos para la formulación y ejecución de los planes estratégicos y operativos de los Planes de Atención Básica-PAB 2004-2007, se considera la vigilancia en salud pública como una función esencial de la salud pública. En esta normativa las líneas estratégicas para el desarrollo de esta función a nivel departamental se orientan al fortalecimiento de la capacidad técnica y operativa que sirva de soporte al desarrollo del sistema en todos los niveles municipales e incluye, entre otros, el desarrollo tecnológico de los Sistemas de Información en Salud y la red de comunicaciones.

Sivigila es un programa de obligatorio cumplimiento en todos los departamentos del país, gestionado por las direcciones departamentales y secretarías de salud. Se encarga de la recopilación de datos, análisis, interpretación y comunicación de información sobre eventos para su uso esencial en la orientación de la toma de decisiones en salud pública. Para el Sivigila los eventos de salud corresponden al conjunto de enfermedades, lesiones de causa externa, causas de muerte, factores protectores, factores de riesgo, acciones de protección es-

pecífica y de detección temprana de enfermedades, muestras biológicas o químicas y demás determinantes asociados.

Después estudiar el funcionamiento del Sistema de Vigilancia en Salud Pública del departamento de Cauca, en el marco de algunos proyectos de investigación y desarrollo del grupo de Ingeniería Telemática de la Universidad del Cauca [2][3]; y sustentados también en el diagnóstico de la situación de los Sistemas de Vigilancia Epidemiológica en Perú realizado en el marco de una tesis doctoral [4]; se identificaron los dos principales problemas de los Sistemas de Vigilancia Epidemiológica con relación al tratamiento de la información:

1. La información existente sobre la situación de salud de la población es incompleta, poco fiable o simplemente no existe.
2. La información existente no se usa adecuadamente en la planeación, ejecución y evaluación de los programas de salud.

Con relación a la confiabilidad de la información, según el informe consolidado del programa Sivigila de diciembre de 2004,¹ en Colombia se reportó información sólo del 90,2% de las unidades notificadoras, y una constante fue el no envío de informes por parte del Chocó, departamento con las mayores dificultades de comunicación y problemas de salud. A nivel local el departamento del Cauca reportó información del 86,6% de las unidades notificadoras, según el

1 Programa Sivigila. Datos tomados del último reporte publicado de Vigilancia Epidemiológica correspondiente a la Semana Epidemiológica del 7-13 de diciembre de 2004. Consultado septiembre de 2004. <http://www.col.ops-oms.org/sivigila/IndiceBoletines2004.asp>

mismo reporte. En este departamento existen municipios que no notifican, ya sea porque son zonas de difícil acceso para recolectar los datos, o porque no existen medios de comunicación confiables (normalmente se hace vía telefónica). Se presume además que gran parte de la información existente es incompleta o poco confiable debido a que en muchos departamentos se recolecta y procesa manualmente, no se almacena en ningún medio y los datos de cada semana se sobrescriben a los anteriores. Adicionalmente, en muchos municipios la notificación semanal no es asumida por las unidades notificadoras responsables sin que exista un control al respecto.

Con relación al problema de información existente que no se usa adecuadamente para ejecutar acciones en salud pública, puede demostrarse que la información recolectada no es devuelta a los respectivos municipios para que se obtenga un perfil epidemiológico de las zonas. En el portal del Instituto Nacional de Salud se publican los informes nacionales consolidados únicamente por departamento, además que no se publican inmediatamente sino unos meses después de la semana de ocurrencia como puede verificarse en la fecha del último reporte consolidado.² En el departamento del Cauca existen otros programas de salud pública que de alguna forma ejecutan funciones de Vigilancia Epidemiológica (Control de la Tuberculosis, Enfermedades

transmitidas por vectores, programa AIEPI, etc.), pero no se tiene integrada esta comunicación dentro del sistema de información para vigilancia epidemiológica del departamento.

Existe entonces la necesidad de contar con un sistema de información que soporte los procesos de vigilancia en salud pública: *recolección sistemática y continua y procesamiento de datos en salud para su posterior conversión en Información (análisis e interpretación de datos) y entrega a los tomadores de decisiones en salud.*

En este artículo se describe en detalle una herramienta desarrollada para soportar los procesos de notificación obligatoria de eventos de Sivigila. Esta herramienta ha sido probada en una versión inicial en el marco del proyecto EHAS-Enlace Hispanoamericano de Salud² donde se implantó un prototipo en cuatro puestos de salud rurales conectados a un centro de salud en el municipio de Silvia, departamento del Cauca, en Colombia [5]. Actualmente se ha desarrollado una nueva versión de la herramienta más versátil, en el marco del proyecto EHAS-@LIS³ financiado por la Unión Europea. Esta nueva versión se diseñó para cumplir las características de reusabilidad, desarrollo basado en componentes, flexibilidad, interoperabilidad y utilización de tecnologías abiertas. La descripción de la arquitectura de este sistema y los componentes implementados son el objetivo de este artículo.

² <http://www.ahas.org>

³ Programa @LIS: Alliance for the Information Society / Alianza para la Sociedad de la Información.
<http://www.europa.eu.int/comm/europeaid/projects/alisis/>

2. MÉTODOS

Para desarrollar sistemas software con características de reusabilidad, desarrollo basado en componentes, flexibilidad e interoperabilidad es necesario diseñar primero una arquitectura que cumpla estos requisitos. En este trabajo se hace uso de las metodologías, marcos de desarrollo, métodos, modelos, técnicas y herramientas más ampliamente usadas en el dominio de la ingeniería de software para el diseño de sistemas con estas características. En la aproximación, la especificación del sistema se hace de arriba hacia abajo, es decir, iniciando con un nivel de abstracción alto de la arquitectura, hasta llegar a un nivel más detallado donde se definen cada uno de los componentes de la arquitectura (desarrollo basado en Componentes). Para obtener el nivel de abstracción más alto y por lo tanto una visión más completa del sistema de vigilancia, se utilizó como referencia el modelo abierto OSI para el Procesamiento Distribuido Abierto (Reference Model for Object Distributed Process RM-ODP [6]). Este marco de referencia provee una visión más amplia *-mediante diferentes puntos de vista incluyendo el de la organización objetivo-* de los Sistemas de Información distribuidos.

Sin embargo, RM-ODP no define un lenguaje propio ni un proceso para el diseño de los componentes del sistema distribuido. RM-ODP debe estar soportado en un proceso de desarrollo para lo cual se utilizará la Arquitectura Basada en Modelos (en inglés Model Driven Architecture MDA). MDA fue seleccionada por ser una tecnología prometedora que asegura la reusabilidad de la arquitectura y el

uso de procesos automatizados para convertir modelos independientes de la plataforma en código fuente [7]. La separación en modelos de MDA la hacen una muy buena aproximación para articular los modelos resultantes del análisis por vistas de RM-ODP. MDA separa entre un Modelo Independiente de la Plataforma (Platform Independent Model, PIM) que será la arquitectura de referencia deseada y el modelo dependiente de la plataforma (Platform Specific Model, PSM), que viene a ser la arquitectura de un sistema final desarrollado a partir de la arquitectura de referencia. Al igual que RM-ODP, MDA en sí misma no es una metodología de desarrollo software. La especificación de MDA sólo proporciona una estrategia general a seguir en el desarrollo de software, pero no define ni técnicas a utilizar, ni fases del proceso, ni ningún tipo de guía metodológica. Para suplir esta deficiencia fue necesario buscar una metodología de desarrollo software basada en MDA para obtener el nivel de detalle (granularidad) más alto en cada una de las vistas anteriores de RM-ODP.

Actualmente la mejor opción metodológica es el Proceso Unificado de Desarrollo de Rational (Rational Unified Process, RUP) pues se ha convertido en el estándar de facto en el desarrollo de software [8]. RUP no incorpora explícitamente el proceso definido por MDA; sin embargo, dado que RUP es en general basado en modelos y prescribe diferentes niveles de abstracción, RUP y MDA son compatibles por naturaleza. A pesar de esto también es cierto que, en términos prácticos, RUP no provee una guía específica de cómo integrar el proceso de desarrollo software de MDA en

el conjunto de procesos RUP. Una justificación para esto, según Brown y Conallen [9], es que RUP refleja las mejores prácticas de la industria de software y normalmente no adopta prácticas novedosas hasta que están completamente aceptadas en el mercado. MDA está aún emergiendo, lo mismo que las experiencias exitosas de aplicar MDA a RUP.

La combinación de estas tres tecnologías (RM-ODP, MDA y RUP) se presenta en la Figura 1. El objetivo de este artículo no es describir el proceso de desarrollo de la arquitectura ni cada uno de los modelos obtenidos. Sólo se describirá el último modelo, el de Implementación, que es el resultado de seguir todo el proceso de diseño propuesto en la metodología. La descripción detallada de las fases de la metodología de desarrollo y los modelos obtenidos puede encontrarse en la tesis de maestría del autor [10]. De igual forma, el artículo “Un Modelo de Referencia para la Especificación y

Análisis de Sistemas de Información para Vigilancia en Salud Pública”, incluido en este mismo número de la revista, describe el modelo de referencia (vista de la organización) usado para la arquitectura y descrito en el modelo de contexto, modelo de comportamiento y modelo conceptual descrito en la Figura 1.

3. RESULTADOS: EL MODELO DE IMPLEMENTACIÓN

En la metodología propuesta el Modelo de Implementación se representa mediante un diagrama de componentes.

La Figura 2 representa el diagrama de componentes de la aplicación. La información es recolectada por la Herramienta Cliente (aplicación Java) en los puestos de salud o los hospitales municipales y enviada por correo electrónico a la Herramienta Servidor (aplicación Web) ubicada en la Unidad Notificadora del municipio para obtener los consolidados de los

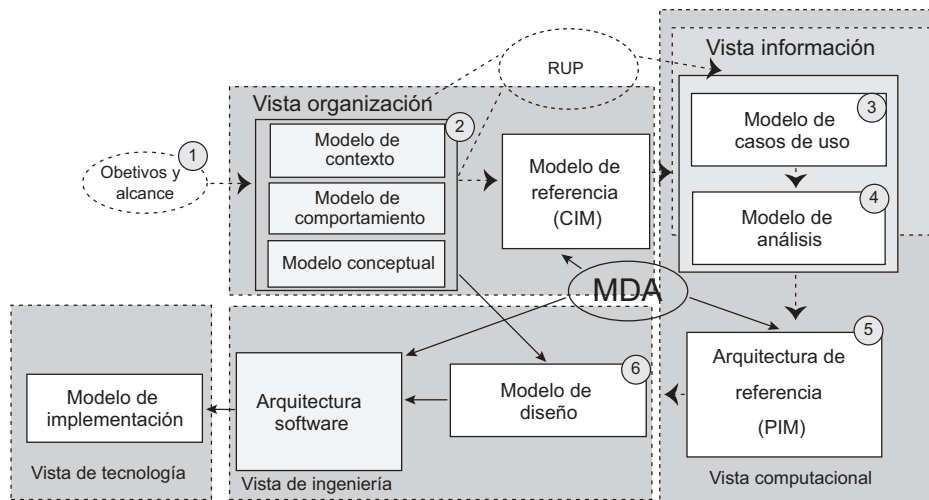


Figura 1. La metodología de Desarrollo.

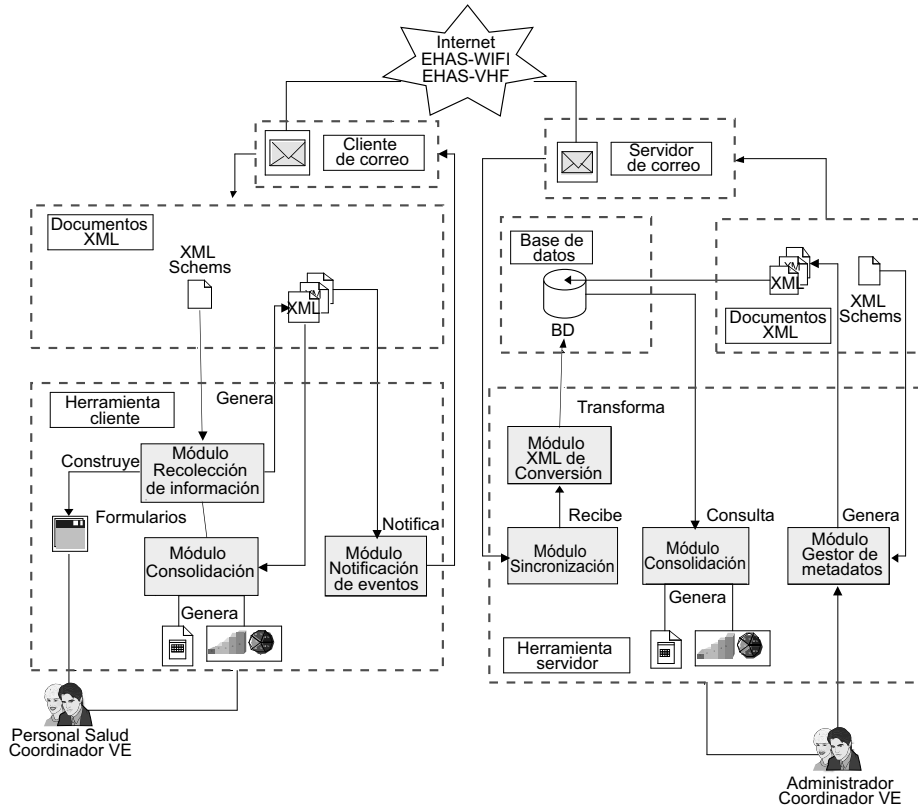


Figura 2. Diagrama de componentes de la aplicación.

eventos epidemiológicos por municipio y ser analizados por el personal de salud encargado. El sistema cuenta con una herramienta gestor de formularios o metadatos, basada en un editor XML, esta herramienta se integró a la aplicación Web, para que el personal de salud encargado pueda crear y modificar sus propios formularios de recolección de información epidemiológica definiendo únicamente el nombre de los campos y el tipo de datos a recolectar ya que la estructura de estos se encuentra predefinida en el editor XML por medio de documentos XMLSchema.

El documento XML generado por el gestor de formularios es enviado a través de la red de comunicaciones, por medio de correo electrónico, a la herramienta cliente en cada puesto de salud del municipio para generar dinámicamente el formulario de recolección. A continuación cada uno de los seis componentes básicos del diagrama son descritos en detalle.

3.1. Componente de recolección de información

Este componente cumple la función de recolección de información epidemiológica y de pacientes registrados

por medio de los siguientes formularios:

- **Formulario pacientes:** Es el que recoge la información de los pacientes que han presentado algún caso de notificación obligatoria, con el fin de tener un registro de los datos personales de cada paciente.
- **Formularios Sivigila:** Son aquellos que siguen el formato del Sistema de Vigilancia Epidemiológica de Colombia, Sivigila. Los formatos son : Enfermedades de notificación obligatoria, Muerte Materno Perinatal y EDA-Enfermedades Diarreicas Agudas/IRA-Enfermedades Respiratorias Agudas.
- **Formularios remotos:** Son aquellos que pueden ser editados desde la Herramienta Servidor para la recolección de información epidemiológica adicional requerida por personal de salud local o los COVE. Estos formularios le permiten al sistema adaptarse a cualquier formato de recolección de información.

La Herramienta Cliente es una aplicación desarrollada en Java y utiliza documentos XML como repositorio de datos y como fuente de datos. La función de *repositorio* es para el almacenamiento de la información recolectada, la función de *fuentes* se realiza cuando el sistema requiere información para la presentación de los formularios, la cual es enviada desde la Herramienta Servidor en documentos XML para la actualización de los formularios, por ejemplo Enfermedades de Notificación Obligatoria (ENO), Municipios, IPS. Para los formularios remotos se utilizan

XmlSchemas como metadatos (estructura de información).

El usuario introduce la información epidemiológica o de pacientes a través de las interfaces *GUI_Formulario* implementadas en Swing y controladas por JavaBeans, esta información es guardada localmente en documentos XML llamados *XML_Escritura*.

La información que se presenta como menú de selección en los *GUI_Formularios* es obtenida de documentos XML llamados de *XML_Lectura*, los cuales son enviados por correo electrónico desde la Herramienta Servidor y se actualizan automáticamente. Los *XML_Lectura* están controlados por JavaBeans que utilizan los API de JDOM para manejo de documentos XML.

Los *GUI_Formularios* son presentados al usuario según una estructura definida previamente en documentos XmlSchemas llamados *Metadatos*. Los Metadatos contienen información sobre título del formulario, nombre del documento XML, nombre de la tabla en la base de datos, nombre de los campos en el formulario, propiedades de los campos (selección o texto), número de campos.

Estos Metadatos son definidos y creados por el Coordinador de Vigilancia Epidemiológica por medio de la Herramienta Servidor y enviados al Cliente por correo electrónico.

IN_Mail: Los documentos *XML_Lectura* y los *XmlSchemas* son enviados desde la Herramienta Servidor por correo electrónico y recibidos por el Cliente. Este proceso se realiza automáticamente, el Cliente se conecta con un servidor de correo predeterminado y busca en él correos por asunto,

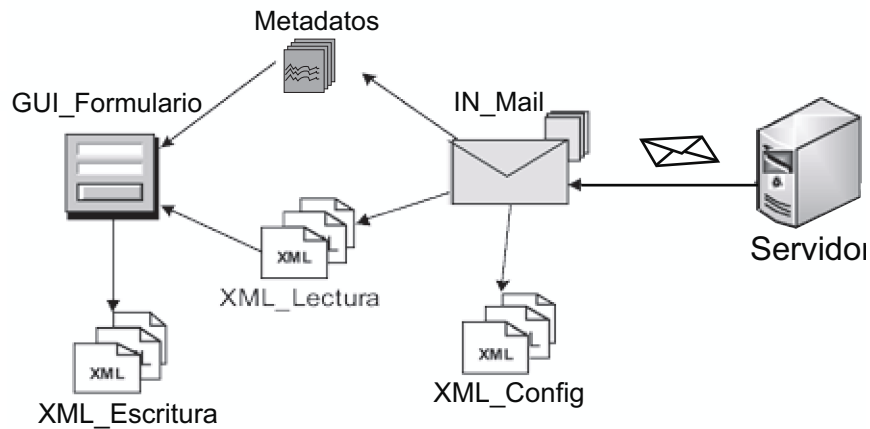


Figura 3. Componente de recolección de información.

en el caso de encontrar un correo realiza la correspondiente comprobación y copia los documentos en una carpeta local para ser accedidos por el sistema Cliente. Se utiliza el API de JavaMail para el procesamiento y manejo del correo electrónico.

Las opciones de configuración para la conexión, envío y recepción de documentos XML a través del servicio de correo electrónico son almacena-

das en un Documento XML llamado *XML_Config*.

3.2. Componente de Notificación de eventos

Este componente cumple la función de Notificación o envío de la información epidemiológica recolectada por el componente de recolección hacia la Herramienta Servidor.

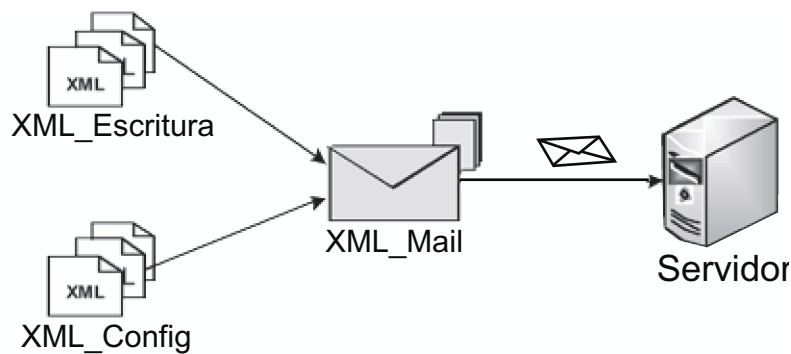


Figura 4. Componente de Notificación de eventos.

XML_Mail: Cuando el usuario lo requiera puede enviar la información epidemiológica recolectada al Servidor por medio de correo electrónico. Los documentos XML_Escritura se envían como archivos adjuntos de un e-mail, sus opciones de envío están almacenadas en un documento XML_Config que es procesado por JavaBeans que utilizan el API JavaMail para manejo de correo electrónico. Los documentos son enviados solamente una vez y se guarda una copia de respaldo localmente, la información no se repite, en cada envío se adjunta

solo la información recolectada desde el ultimo envío. Este proceso se realiza automáticamente por el sistema, de esta forma el usuario no requiere conocimientos en correo electrónico para su utilización.

3.3. Componente de consolidación de información

Este componente cumple la función de consolidar la información en reportes epidemiológicos para ser analizados por el personal de salud local o por los Comité de Vigilancia Epidemiológico (COVE).

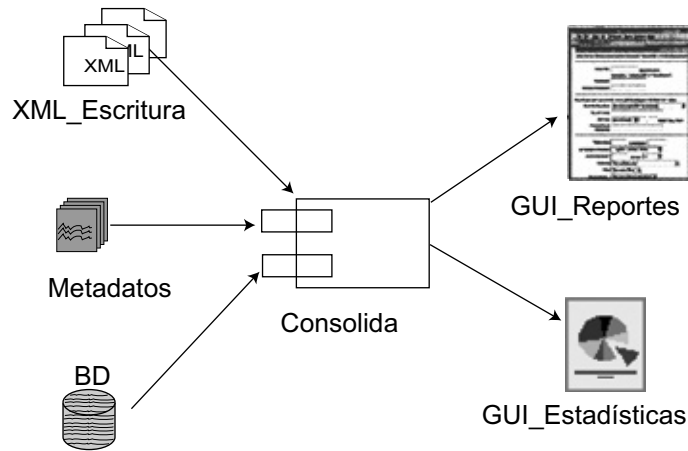


Figura 5. Componente de consolidación de información.

XML_Consolidado: Los consolidados o reportes epidemiológicos son presentados en el Cliente con la información almacenada localmente y en la Herramienta Servidor con la información obtenida de los diferentes Clientes. Para esto se utilizan los Metadatos, los cuales definen los campos que serán incluidos en los consolidados y la información epide-

miológica es obtenida de los XML_Escritura almacenados localmente en el Cliente o en la base de datos de la Herramienta Servidor.

Los *GUI_Consolidado* son las interfaces que presentan los reportes consolidados en el Cliente o en la Herramienta Servidor, a través de ellos el usuario puede desplegar la información epidemiológica recolec-

tada en un periodo determinado para su análisis.

3.4. Componente XML de conversión

En la Herramienta Servidor el componente que cumple la función de convertir la información de lectura almacenada en la base de datos en

documentos XML es *XML_Servlet*. Estos documentos son enviados junto con los Metadatos al Cliente, por medio de correo electrónico. De esta forma la información almacenada en nuestra base de datos será mapeada a documentos XML_Lectura para ser utilizados posteriormente por el Cliente.

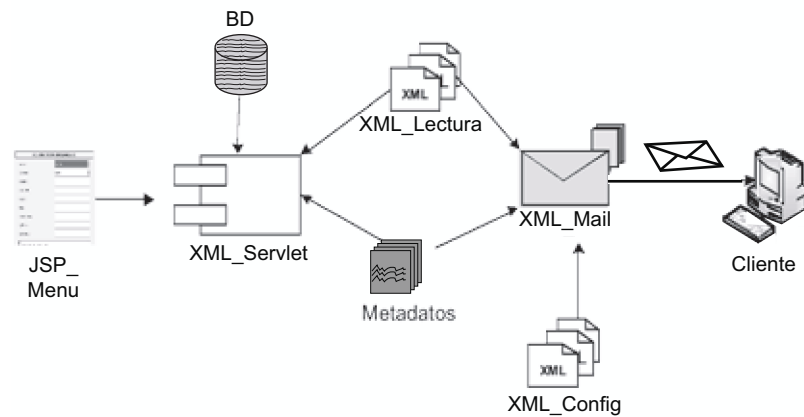


Figura 6. Componente XML de conversión.

3.5. Componente de sincronización

Los documentos XML_Escritura son enviados desde el Cliente por medio de correo electrónico a la Herramienta Servidor; automáticamente el Servidor revisa en su bandeja de entrada los correos según el asunto, si los encuentra guarda los documentos XML adjuntos en una carpeta local, de esta forma se puede realizar alguna función de depuración para verificar que la información que se va a insertar a la base de datos no se encuentre duplicada o con errores.

Los documentos XML_Escritura del Cliente son leídos por el Servidor e insertados a la base de datos, a través del componente XML_Sincroniza, en este proceso se incluyen algunas reglas para evitar errores en la sincronización de la información como la implementación de claves primarias y la comprobación de documentos XML bien formados por medio de los XmlSchemas creados para los Metadatos. Para este componente se utilizan los API de JDOM para manejo de documentos XML y JavaBeans manejados por Servlets para procesamiento de la información en la base de datos.

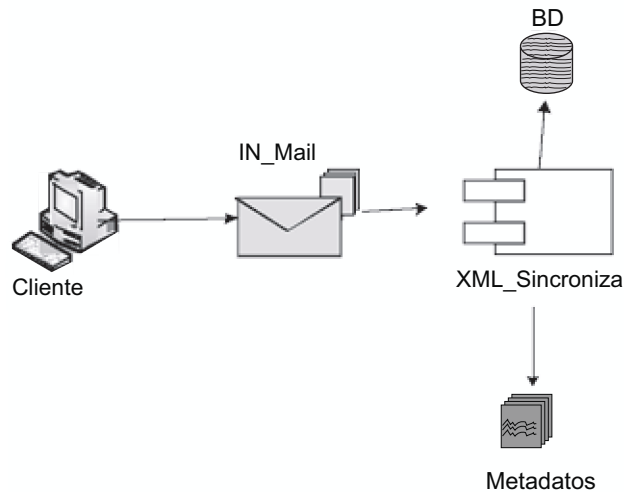


Figura 7. Componente de sincronización.

3.6. Componente Gestor de Metadatos

Este componente es el que se encarga de crear los *Metadatos* o documentos de esquema, los cuales serán utilizados en el sistema para:

- Definir la estructura y propiedades de los formularios remotos que aparecerán en el Cliente. (Título, Nombre de campos, cajas de texto, de selección).
- Definir la estructura de los reportes consolidados en el Cliente y el Servidor. (Campos a desplegar, tipos de datos).
- Verificar que los documentos XML estén bien formados antes de insertarlos a la base de datos del Servidor.

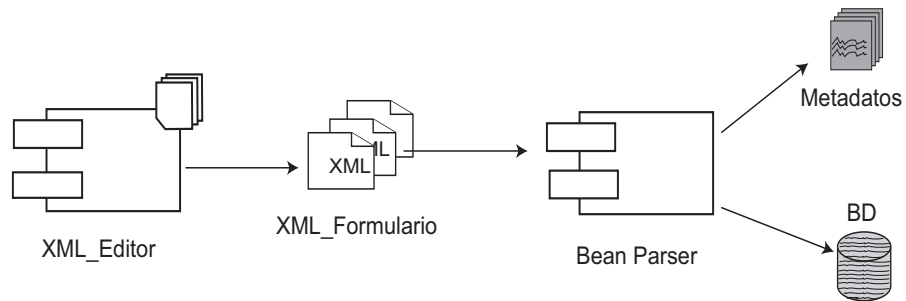


Figura 8. Componente Gestor de Metadatos.

El XML_Editor es una aplicación o herramienta funcional que le permite al usuario editar o crear XML_Formularios para recolección de información epidemiológica de una forma amigable y sencilla en formato XML sin necesidad de saber sobre este lenguaje de marcado. El componente BeanParser se encarga de transformar estos documentos a Metadatos y convertirlos en una tabla para la base de datos de la Herramienta Servidor.

4. DISCUSIÓN

Se describe a continuación cómo se ha planeado técnicamente la implantación de la herramienta descrita anteriormente, así como el piloto probado en un escenario real en el marco del proyecto EHAS-@LIS y soportando el programa Sivigila.

4.1. Implantación técnica

El sistema de información hará uso de la red de comunicaciones de EHAS. La red EHAS consta de enlaces inalámbricos vía WI-FI,⁴ VHF y HF, y presta servicios de correo electrónico e Internet en algunos puntos, se dispone de equipos computacionales en cada establecimiento de salud con los requerimientos mínimos de un PC comercial (Procesador 2 GHz, 128 MB en RAM, Disco duro de 40GB) y sistema operativo Linux. En Colombia se han realizado pruebas con usuarios no reales para el sistema Sivigila que define las enfermedades de notificación obligatoria, obteniendo reportes y gráficas estadísticas inmediatamente, además de un aumento de la

información epidemiológica en casi un 60%.

La arquitectura de la aplicación adaptada para ser usada por el programa de vigilancia de alertas tempranas Sivigila en el departamento del Cauca, en Colombia, se presenta en la Figura 9. En el centro de salud u hospital, el Coordinador de Vigilancia Epidemiológica (VE) rellena semanalmente a través de un formulario electrónico por medio de la Herramienta Cliente, un reporte con los eventos epidemiológicos registrados en su zona. Este reporte viaja hacia la Herramienta Servidor ubicada en la Dirección Regional de Salud del departamento (DDSC) utilizando la Red EHAS. En su destino, el sistema almacena la información de todos los puestos de salud y hospitales de Primer nivel.

(Unidades notificadoras) en una Base de Datos. Con esta información el encargado de la Sección de Vigilancia Epidemiológica genera un reporte consolidado Sivigila que será enviado al Ministerio de Protección Social donde será ingresado al Sistema Nacional de Salud Pública. El Coordinador de VE en la DDSC puede obtener otros reportes más detallados, que le permitan planificar la atención en salud pública de todo el departamento. A nivel local, el coordinador de vigilancia epidemiológica puede consultar la Base de datos de eventos epidemiológicos y generar reportes locales que son usados en los COVE locales.

⁴ WI-FI: Wireless Fidelity, www.wi-fi.org

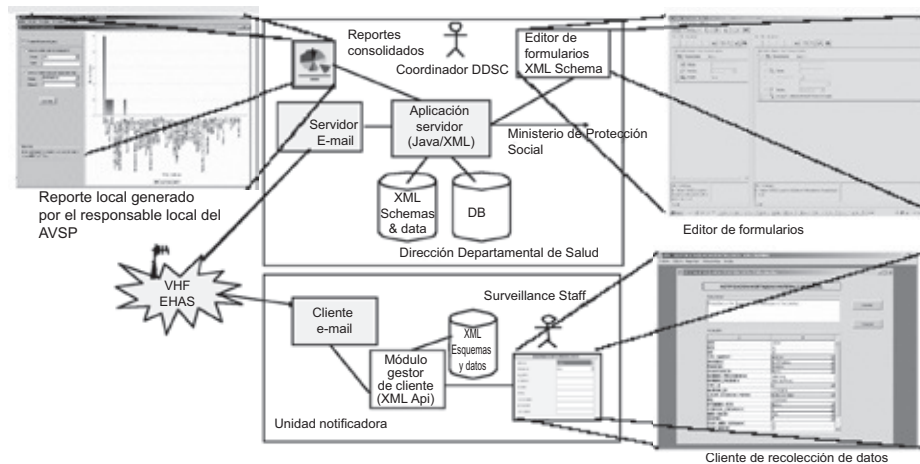


Figura 9. Arquitectura de la aplicación.

Una funcionalidad novedosa fue agregada en las últimas versiones de la aplicación que permite al coordinador de VE en la DDSC diseñar los formularios que se le mostrarán al personal de salud en las unidades notificadoras. Esto es posible porque además de la información rellena en los formularios, se envía a través de la red la estructura de los datos (información de cómo deben aparecer los formularios) almacenada en unos documentos XML de estructura: XMLSchemas. De esta forma el Coordinador de VE en la DDSC diseña el formulario electrónico con un Editor XML (Similar a un editor de páginas Web), y su diseño se actualiza automáticamente en la interfaz de la Herramienta Cliente. De esta forma, si el Ministerio de Protección Social solicita nuevos datos o modifica la estructura de los mismos, este cambio será transparente en la aplicación del Puesto de Salud, además le permite al sistema

adaptarse a los diferentes formatos de recolección de información epidemiológica de cada país.

La aplicación está desarrollada totalmente en herramientas de Software Libre, tecnologías multi-plataforma como Java, Java Server Pages (JSP), Servlets. La Herramienta Cliente es una aplicación de escritorio que usa interfaces Java Swing. La información en el Cliente es almacenada en documentos XML y una estructura mediante XML Esquemas. El API JDOM de Java gestiona los documentos XML. La Herramienta Servidor está desarrollada con Servlets y JSPs que gestionan de igual forma los XMLSchemas y documentos XML mediante el API JDOM. La base de datos está implementada en FireBird, un potente gestor de bases de datos relacional. El editor de formularios ha sido adaptado de un Editor XML en Java de código libre. El editor permite a través de una interfaz gráfica generar formularios electrónicos en

un XML_metadata que describe la estructura de dicho formulario.

4.2. Validación en un escenario real

Para lograr un correcto análisis y evaluación del impacto obtenido con el uso del sistema se requieren por lo menos seis meses de estudio recolectando y generando información epidemiológica con el sistema prototipo, y de otros seis meses para realizar mejoras y adecuaciones de la herramienta según el estudio en cada país con el modelo propuesto. Por dificultades en la realización de las instalaciones y falta de recursos, se está a la espera de obtener financiación adicional para completar la validación.

Para su validación se propone implantar un modelo de preparación, envío, procesado, visualización y automatización de información epidemiológica, a través de correo electrónico, utilizando la herramienta descrita en el presente trabajo. Para su realización es necesario conocer bien los formularios de recolección de información oficiales, los plazos de envío, los modelos de codificación de la información de salud, las responsabilidades de envío y recepción, el uso actual de los reportes consolidados y estadísticos, así como el uso deseado. Después de su implementación debe realizarse un estudio en detalle de las fortalezas y debilidades obtenidas con el uso de la herramienta. Esta actividad requiere también de concienciación del personal de salud a varios niveles jerárquicos y de la promulgación de los resultados y reportes generados.

5. CONCLUSIONES

Se ha mostrado en este artículo cómo puede desarrollarse una aplicación para vigilancia de alertas tempranas con características de reusabilidad, flexibilidad, desarrollo basado en componentes, interoperabilidad y utilización de tecnologías abiertas. El sistema es reusable y flexible pues permite ser adaptado a los diferentes niveles del sistema de vigilancia (municipal, departamental, nacional), a las exigencias de los Sistemas de Vigilancia en Salud Pública (vigilar otros eventos, factores de riesgo, intervenciones, gestión de programas de salud) y a los Sistemas de Vigilancia de otros países Latinoamericanos donde se desarrolla el programa EHAS.

El sistema de información desarrollado permite realimentación de forma que la información generada y los datos agregados puedan volver de manera sencilla a los establecimientos rurales de salud para poner solución a sus problemas epidemiológicos eficazmente. El sistema es basado en componentes tal como son descritos en la sección anterior. Es interoperable porque al usar XML permite que pueda comunicarse con cualquier otro sistema de información en salud utilizando las propiedades de gestión de información que provee esta tecnología y usa tecnologías abiertas como Java, Servlets, JSP, XML.

6. AGRADECIMIENTOS

Este trabajo ha sido realizado gracias al apoyo del Programa AlBan, Programa de becas de alto nivel de la Unión Europea para América Latina, n° de identificación E03D16197CO”, el proyecto EHAS-@LIS contrato ALA/2002/047-639/3151 y la Uni-

versidad del Cauca bajo comisión de estudios otorgada con resolución 136 del 28 de octubre del 2003. Especial agradecimiento al estudiante de maestría Jorge Alexander Figueroa por los aportes al desarrollo de la herramienta aquí descrita.

7. BIBLIOGRAFÍA

- [1] Lineamientos para la formulación y ejecución de los planes estratégicos y operativos del PAB 2004 – 2007 y de los recursos asignados para salud pública. República de Colombia, Ministerio de la Protección Social. Bogotá. Consultado. Enero de 2005. Disponible en: URL: <http://www.paho.org/English/AD/DPC/CD/imci-aiepi.htm>
- [2] Corrales J., Figueroa A., León C. SIGSA – Vigilancia Epidemiológica soportada en un Sistema de Información Geográfica. V Encuentro de Investigación de Tecnologías de la Información Aplicadas a la Solución de Problemas. Medellín: Universidad Nacional de Colombia sede Medellín, 2004. v.1. p.1 – 175
- [3] Figueroa J. A., Vidal M. Modelo para la definición, recolección y entrega de información en un sistema de información para vigilancia epidemiológica. Trabajo de Grado. Facultad de Ingeniería Electrónica y Telecomunicaciones. Universidad del Cauca, Popayán, 2003.
- [4] Martínez A. Evaluación de impacto del uso de tecnologías apropiadas de comunicación para el personal sanitario rural de países en desarrollo. Tesis Doctoral. Universidad Politécnica de Madrid. 2003.
- [5] Martínez A., López D.M., Sáez A., Seoane J., Rendón A., Shoemaker R. G., Fernández I. Improving Epidemiologic Surveillance and Health Promoter Training in Rural Latin America through ICT . *Telemed J E Health*. 2005 Aug;11(4):468-76.
- [6] ISO/IEC 10746-1, 2, 3, 4 | ITU-T Recommendation X.901, X.902, X.903, X.904. “Open Distributed Processing - Reference Model”. OMG, 1995-98.
- [7] Mellor S.J. and Clark A.N. and Futagami, T. Model-driven development – Guest editor’s introduction. *IEEE Software*, 20(5):14-18, Sept.-Oct. 2003.
- [8] Rumbaugh J., Booch G., Jacobson I. “The Unified Modeling Language Reference Manual,” Second Edition, Addison-Wesley, 2004.
- [9] Brown A. W., Conallen J. An introduction to model-driven architecture. Part III: How MDA affects the iterative development process. *IBM Rational*. 2005. Consultado Junio 15 de 2005. Disponible en: <http://www-128.ibm.com/developerworks/rational/library/may05/brown/>
- [10] López D. Una Arquitectura de Sistemas de Información para vigilancia en salud pública en zonas rurales. Tesis de Maestría. Universidad del Cauca. 2005.

8. CURRÍCULO

Diego M. López. Ingeniero en Electrónica y Telecomunicaciones y Magíster en Ingeniería con énfasis en Telemática de la Uni-

versidad del Cauca. Actualmente cursa sus estudios de doctorado en Informática para la salud en el Health Competence Center de la Universidad de Regensburg en Alemania. Es profesor asociado del Departamento de Telemática de la Universidad del Cauca. Cuenta con más de 30 publica-

ciones en revistas y congresos, especialmente en el área de la Salud. Sus áreas de interés son estándares de interoperabilidad en salud, interoperabilidad semántica, arquitecturas de sistemas de información, sistemas distribuidos y sistemas de información en salud pública. ☀