

## SIRIO, UN EDITOR DE INFORMES CLINICOS

*García Mataix, Ricardo - Técnico de Sistemas de Hospital La Fe de Valencia*  
*Santisteban Navarro, Asunción - Analista de Aplicaciones del Hospital La Fe de Valencia*

### INTRODUCCIÓN

En un hospital se escriben multitud de informes clínicos con estructuras muy diferenciadas. Se hacía necesario ordenar y estandarizar la gran variedad de informes distintos en base a un modelo homogéneo. La solución, bautizada con el nombre de SIRIO, debía ser suficientemente versátil, como para conseguir que el propio facultativo se involucrara directamente en el proyecto. Para todo ello, se ha desarrollado un modelo jerarquizado de informes, un editor asistido para la redacción del informe y un visor de información que accede a toda la información clínica accesible actualmente en el hospital.

El proyecto se encuentra ya en fase de explotación, siendo el informe de alta de Urgencias el primero que se ha desarrollado en base a esta arquitectura.

Al facultativo se le atrae al proyecto desde una doble perspectiva. En primer lugar se muestra la sencillez del manejo del editor y las ventajas de tenerlos almacenados en una base de datos documental, tanto por su gran accesibilidad, como para estudios de casos clínicos con una complejidad mayor de la que disponemos actualmente. Por otro lado, se le ofrece, integrado en el editor, un acceso a toda la información de la que disponemos del paciente en los distintos subsistemas del hospital.

La aplicación debía de tener una interfaz de usuario muy rica. Al mismo tiempo, un gran número de instalaciones habrán de sumarse al proyecto (en fase muy inicial, tenemos ya más de 20 puestos en funcionamiento), pero todos ellos dentro de una Intranet. La recomendación J2EE para este tipo de arquitecturas parece clara: Interfaces nativas, pero con un mecanismo de actualización automática de versiones. Hemos utilizado utilizado clientes Java para el desarrollo de interfaz de usuario y Java-WebStart como protocolo JNLP de actualización automática de versiones.

### DISCUSIÓN

Java ha sido el entorno utilizado para desarrollar todo el proyecto. El carácter 100% orientado a objetos del lenguaje, así como su inmensa API pueden posibilitar un entorno integrado que permita funcionalidades tan diferenciadas como conexión a múltiples bases de datos para la obtención de la información clínica de los distintos subsistemas del hospital (HIS, RIS, Anatomía Patológica y -en breve- Laboratorios), como la construcción de un procesador de textos completamente orientado a la edición de un informe clínico, como el modelado jerárquico de los distintos tipos de modelos de informes en base a XML.

### Visor de la Información clínica

Accedemos a la información administrativo-clínica de un paciente, a su historial radiológico (podemos ver ya también las imágenes almacenadas en un PACS), a los informes de anatomía patológica. Estamos a las puertas de conectar a este sistema también los informes de los diferentes Laboratorios de nuestro hospital. Cada una de las conexiones a los distintos subsistemas del hospital se realiza en un hilo de ejecución diferente, evitándose así los atascos que supone siempre una recopilación monoentrelazada a partir de sistemas tan heterogéneos.

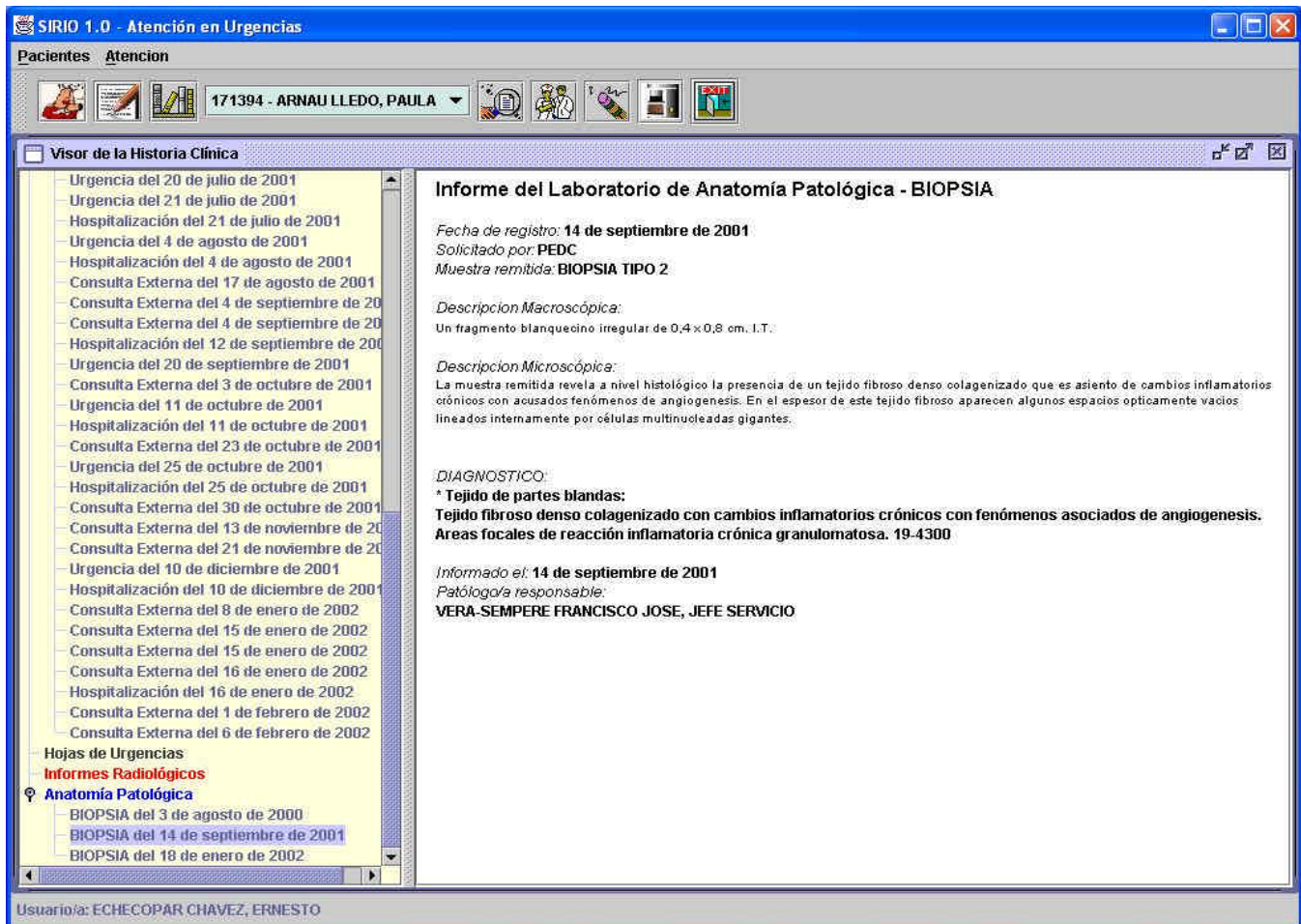


figura 1

**Sirio, el editor clínico**

El editor clínico está pensado para satisfacer tanto las necesidades de facilidad de uso que exige un facultativo, como las necesidades de estructuración de la información que exige un sistema de información. Como primera aproximación, un documento consta de una cabecera y de un conjunto de apartados. La escritura del documento presenta las facilidades estándar de edición (teclas rápidas, copiar y pegar). Asociados a cada uno de los apartados del documento tenemos los ayudadores específicos. Estos ayudadores aparecen automáticamente al entrar en el apartado correspondiente. Al facultativo se le explica que mediante estos ayudadores la confección del documento es mucho más rápida y estandarizada. El uso de los ayudadores, sin embargo, va más allá que su utilización como macros fácilmente accesibles. Si bien, el usuario percibe, que el texto asociado a un ítem del ayudador, en los interiores de la aplicación han ocurrido otras cosas. El ayudador pasa la información en cuestión a una clase que hemos llamado el modelo del documento.

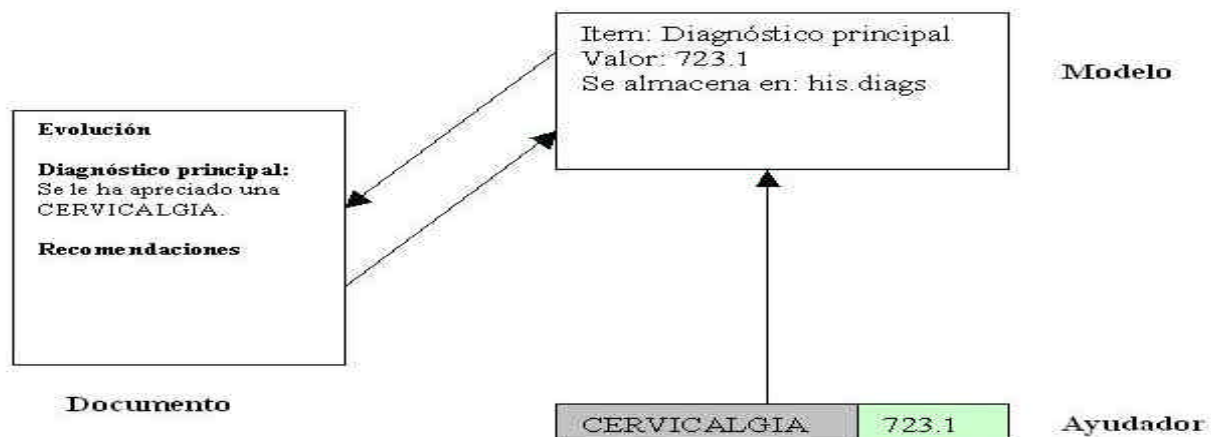


figura 2

Esta clase está al tanto de muchas cosas: almacena la codificación asociada a los descriptores elegidos, mantiene la integridad del documento, conoce las restricciones (p.ej, debe existir un diagnóstico principal y sólo uno) y conoce exactamente cómo almacenar toda esta estructura de datos en tablas correspondientes. El modelo pasa al documento el ítem de información. El documento mantiene también una referencia a estos ítems y conoce en qué lugar físico del texto se encuentran posicionados. De esta forma, el documento sabe qué partes del texto se han de tratar de forma conjunta a la hora de eliminar. Cuando ello ocurre, el documento avisa a su vez al modelo de esa contingencia.

SIRIO guarda también el documento en formato rtf en una base de datos documental, con posibilidad de búsquedas indizadas sobre cualquier parte del texto (Informix 9 y el blade Excalibur).

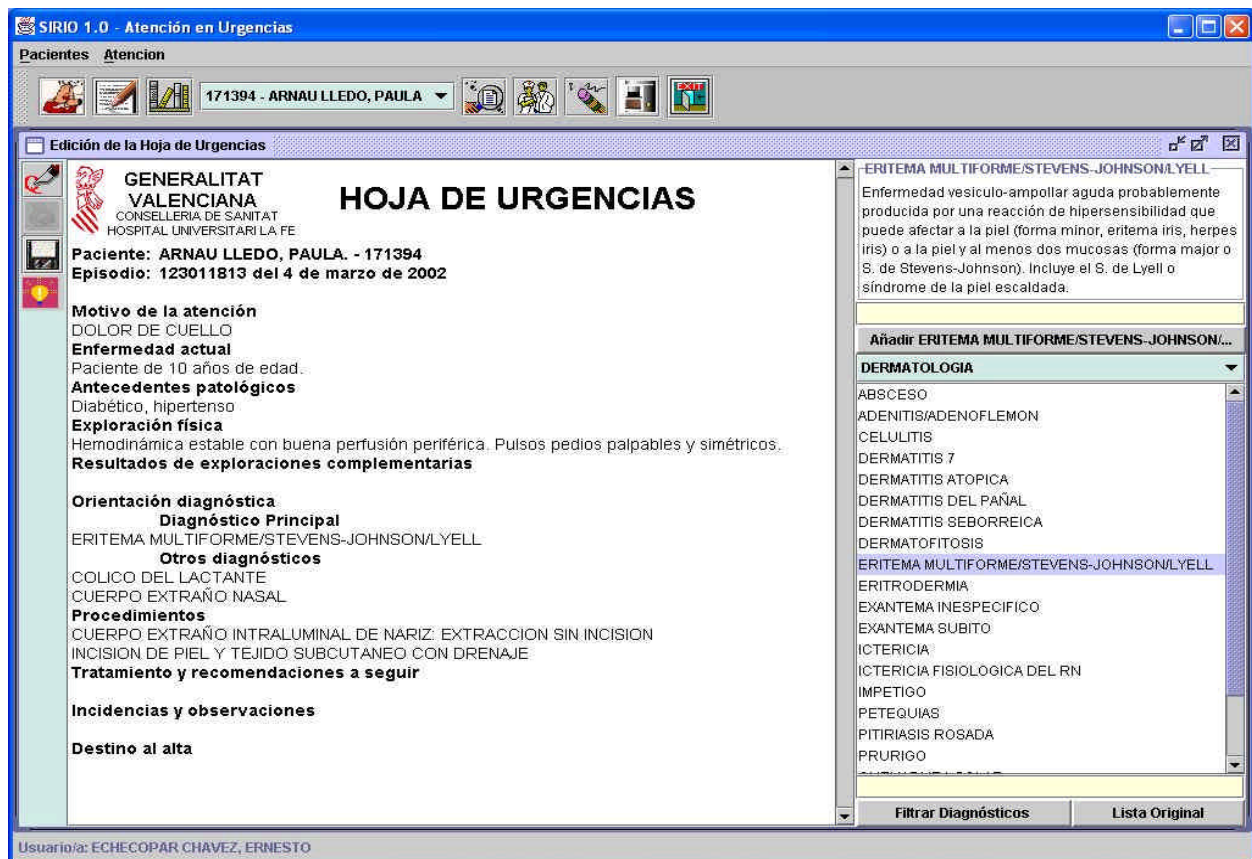


figura 3

Las clases más importantes que intervienen en el diseño de este sistema son las siguientes. Un informe, consta de un modelo, un documento y de un editor.

Los distintos tipos de informe se organizan de forma jerárquica, así por ejemplo un informe de alta de urgencias es un tipo de informe, toda la funcionalidad ya definida en el informe genérico la tenemos a nuestra disposición.

Ello significa que para definir un tipo de informe distinto, sólo necesitamos sus apartados, sus ayudadores y su modelo específico, toda la funcionalidad y complejidad para ofrecer una interfaz de usuario que posibilite este sistema ya la heredamos gratis de las clases genéricas.

De esta forma, también tenemos definido ya lo que será un informe de alta de hospitalización genérico, clase derivada de informe y que servirá a su vez como clase antecesora de informes de alta más especializados en función de la especialidad médica de que se trate.

Pensamos que esta forma de proceder es la que impone un orden estructural en el caos de papeles que circulan en un sistema tan complejo como un hospital, sin perder por ello riqueza semántica tan necesaria en nuestro entorno.

## **Perspectivas**

Las futuras versiones de SIRO se van a encaminar en tres caminos distintos: En primer lugar, la extensión del editor, para la redacción de los informes de alta de hospitalización. El acceso on-line a toda esta documentación será un paso importante en la dirección de una historia clínica informatizada. En segundo lugar, queremos rediseñar SIRIO, para poder especificar toda la estructura de un tipo de documento concreto en un documento XML. SIRIO será capaz de construir dinámicamente el editor adecuado en base a la especificación en tiempo de ejecución.

En tercer lugar, estamos trabajando en la línea de colocar una capa Web-Services (implementada en Java en nuestro caso) por encima de cada uno de los subsistemas que participan en la información clínica. Cada subsistema ofrece al exterior la información que se defina como un servicio Web. Ello posibilitará la reescritura de SIRIO y sustituir los accesos directos a las bases de datos, con una capa de interoperabilidad basada en los últimos estándares del mercado. Ello redundará en la modularidad del sistema, liberando a SIRIO del conocimiento de las implementaciones de cada uno de los subsistemas del hospital. Por otro lado, el acceso a nuestras aplicaciones mediante servicios Web, libera completamente a la aplicación cliente de la utilización de ninguna plataforma de implementación.