

Integración de Bases de datos hospitalarias:

Gestión de las bases de datos Componentes

Pere Crespo (pedcremo@inf.upv.es), Jose Alberto Maldonado (jamaldo@upvnet.upv.es), César Cano (cecano@fis.upv.es), Montserrat Robles (mrobles@fis.upv.es), Juan Carlos Casanova (casanova_jca@gva.es)

Grupo de Informática Médica, E.U.I. Universidad Politécnica de Valencia, Valencia 46022

Abstract La aplicación visual que se describe en el presente artículo forma parte de un proyecto cuyo objetivo es el desarrollo de un sistema informático que permita a los profesionales sanitarios acceder, de manera controlada, a la información sanitaria sobre los pacientes, que se encuentra repartida en múltiples sistemas de información heterogéneos y autónomos dentro de un hospital. El sistema está basado en el estándar de arquitectura de historia clínica electrónica ENV13606 del CEN/TC251, el cual se utiliza como modelo canónico para la representación de la información clínica.

Esta herramienta está pensada y diseñada para ser utilizada por personal informático cualificado de un hospital. Estos usuarios deben conocer a fondo los sistemas de información departamentales del centro, las bases de datos clave de la organización, que aplicaciones las utilizan, que relaciones existen entre ellas, qué información es relevante sobre los pacientes etc. Con toda esta información y con esta aplicación visual la persona o personas encargadas del S.I hospitalario pueden llevar a cabo las siguientes tareas: Importar esquemas BBDD, representación en un formato unificado, definir información demográfica básica para la identificación de un paciente, completar los esquemas (permitiendo la adición de nuevas dependencias de equivalencia entre conjuntos de atributos, claves ajenas), definir qué información puede ser compartida con el resto de usuarios, determinar automáticamente qué información es relevante para la historia clínica o sea tablas que contienen información puramente sanitaria, realizar consultas SQL ...

La aplicación forma parte de un conjunto de software que será utilizado en los hospitales, estas organizaciones cuentan con sistemas informáticos muy heterogéneos por esta razón el "Gestor" ha sido implementado utilizando JAVA un lenguaje de programación moderno, orientado a objetos, independiente de la plataforma y de propósito general.

Introducción

El objetivo del proyecto dentro del cual se sitúa la aplicación descrita en este artículo es la integración de las bases de datos de una entidad hospitalaria con el propósito de que los profesionales sanitarios puedan acceder de una manera controlada, a la información sanitaria sobre los pacientes.

Nuestra solución se basa en el desarrollo de un servidor que se sitúa entre los usuarios finales y las bases de datos ya existentes y que por tanto se puede considerar como una forma de middleware. A través de este servidor los profesionales sanitarios pueden obtener toda o parte de la información sanitaria sobre un paciente integrada en un formato unificado. Por tanto, en esta solución hacemos más hincapié en la compartición de datos entre los sistemas más que en la integración de los sistemas que contienen los datos. De esta forma, el servidor permite la gestión de una historia médica electrónica "virtual". Decimos virtual porque los usuarios finales pueden disponer de una historia médica electrónica de los pacientes, pero realmente ésta no existe como entidad propia en ningún sistema informático sino que se encuentra repartida por múltiples bases de datos. La historia clínica, o parte de ésta, se construye "al vuelo" y siempre bajo demanda de los usuarios finales. Obviamente esta solución implica la utilización de una estructura común para representar la información sanitaria: arquitectura de historia clínica electrónica. Una arquitectura de historia clínica electrónica modela las características genéricas aplicables a cualquier anotación en una historia clínica independientemente de la organización (primaria, especializada, etc). La arquitectura debe proporcionar principalmente constructores o mecanismos para capturar fielmente el significado original de la información y asegurar que la historia clínica sea comunicable. Se entiende por comunicable que la comunicación de partes de la historia clínica entre diversos profesionales u organizaciones sea segura, y que, por tanto, se salvaguarde el significado original de los datos.

Así, hemos utilizado el estándar de arquitectura de historia clínica desarrollada por el comité técnico TC251 del Comité Europeo de Normalización (CEN) conocida como prENV13606. La razón de esta elección es

simple, este estándar va en camino de convertirse en el estándar definitivo en cuanto a arquitectura de historia clínica en Europa lo cual supone una base firme, y aunque el proyecto no tiene como objetivo el desarrollo de una historia clínica electrónica, sino que se queda un paso atrás (integración a nivel de una organización), consideramos que la estrategia elegida es compatible y facilitará futuros desarrollos o implementaciones de sistemas de historia clínica electrónica.

Los profesionales sanitarios suelen manejar un conjunto más o menos fijo de documentos o agregados de información para la realización de sus actividades (por ejemplo, informe de alta, hoja de solicitud de ingreso, hoja de urgencias, hoja de evolución, etc.). Por tanto, es conveniente que los usuarios finales puedan realizar peticiones de información sobre los pacientes por medio de estos documentos o agregados o cualquier otro definido a posteriori. Estos documentos o agregados pueden ser representados fácilmente utilizando las estructuras definidas por el estándar. Esto nos lleva a un conjunto de lo que nosotros denominamos arquetipos o plantillas, que ya poseen un significado clínico, y que extienden las clases definidas en ENv13606. Estas clases clínicas pueden “enlazarse” fácilmente a los esquemas de las bases de datos a integrar y los usuarios finales realizan peticiones de información por medio de una o varias instancias de estas clases.

Es por tanto necesario definir con qué mecanismos o herramientas se podrá decidir que bases de datos van a formar parte del dominio sobre el cual se realizarán peticiones de información. A su vez, dentro de cada una de las bases de datos a integrar también es necesario definir que tablas y campos de éstas contienen información relevante sobre la historia electrónica de los pacientes, propósito último y final del sistema en general.

Para el desarrollo de herramientas que permitan definir que información es relevante para la historia clínica de un paciente hay que tener en cuenta los siguientes problemas:

- Es difícil decidir qué bases de datos y a su vez que tablas y campos dentro de estas van a ser relevantes para la HCE si no se conocen mínimamente los diseños de los esquemas y que información contienen cada una de las bases de datos y tablas.
- Normalmente el conjunto de bases de datos que pueden coexistir en un hospital puede ser muy heterogéneo en cuanto a fabricantes y tipo del sistema de bases de datos.
- Es habitual que muchas bases de datos no contengan las relaciones adecuadas entre las diferentes tablas que la componen ya sea por mal diseño de los esquemas o por cuestiones de compatibilidad de las sucesivas versiones en las que ha evolucionado la BBDD y las aplicaciones que las utilizan.

Objetivo

La solución propuesta para este problema de gestión de esquemas se ha plasmado en la herramienta visual, “*Gestor de esquemas de bases de datos*”. Esta herramienta nos permite importar los esquemas de las BBDD que vayamos a necesitar y poder tener de una manera integrada una misma visión de los esquemas de las diferentes BBDD implicadas, independientemente de su tipo. Para poder importar las bases de datos el gestor utiliza la API JDBC (Java DataBase Connectivity) de Java que permite extraer el esquema de cualquier base de datos para la cual se tenga el correspondiente driver JDBC, actualmente existen un gran número de drivers para los mas diversos y conocidos sistemas gestores de bases de datos. De esta manera superamos el problema de la heterogeneidad de las BBDD hospitalarias.

Cuando ya hemos importado las BBDD necesarias estamos en un punto en el que aún no hemos hecho ninguna selección de que información va a ser relevante, el gestor provee mecanismos digamos, inteligentes para ayudar en este proceso. El mas importante es el hecho de que se pueda elegir que tabla o tablas de una BBDD van a ser las que contengan la información demográfica o de identificación unívoca más importante sobre los pacientes. Con esta elección el sistema decide que tablas están relacionadas de manera directa o indirecta a través de claves ajenas con la tabla principal que llamamos “**tabla identificadora de paciente**”, dejando aisladas aquellas tablas que no pueden alcanzar a ésta a través de ninguna relación, ya sea directa o indirectamente. Además, el usuario tiene la posibilidad de cambiar manualmente la disponibilidad de aquellas tablas o campos que considere oportunos.

El problema de la falta de claves ajenas de una base de datos, ya sea por cuestiones de un mal diseño o difícil mantenimiento de las aplicaciones que luego las utilizan, se soluciona incorporando al sistema una herramienta para definir relaciones de claves ajenas virtuales entre tablas. Estas relaciones solo existen en el *gestor de esquemas*, en ningún caso afectan a los esquemas originales de cada una de las bases de datos. Es conveniente tener en cuenta que todo el sistema, no sólo el gestor, va a ser una herramienta de sólo lectura, en ningún momento se va a poder alterar ni los datos contenidos en las BBDD originales, ni sus esquemas, ni impedir que las aplicaciones que utilizaban dichas BBDD sigan funcionando de manera habitual. Hay que tener en cuenta que la generación de las consultas SQL que permiten instanciar las plantillas se generan de forma semi-automática, de ahí la importancia de la existencia de relaciones de clave ajena (virtuales o reales) para hacer posible y facilitar la creación automática de estas consultas.

Finalmente una vez importadas las bases de datos que vamos a utilizar y definida que información va a estar disponible para la definición de los arquetipos de la HCE lo que tenemos es un conjunto de BBDD disponibles para posteriormente poder establecer relaciones entre los atributos de los arquetipos y los campos disponibles de estas bases de datos. De alguna manera los arquetipos que definimos posteriormente en tiempo de ejecución con el "Editor de arquetipos" son pequeños esquemas que integran y proveen una vista unificada que puede ir creciendo ya que algunos componentes contienen a otros, de hecho el componente raíz del estándar puede agrupar a todos los demás. Básicamente los arquetipos, según la teoría de bases de datos, no son más que esquemas federados en una base de datos federada débilmente acoplada, ya que no existe un esquema global. Esta herramienta que aquí se presenta permite, entre otras funciones, definir qué información de las bases de datos componentes se hace accesible a la federación.

Metodología

Es una aplicación basada en el típico esquema de comunicación cliente-servidor. En la parte servidora tenemos el **servidor de metainformación** que gestiona el **diccionario de datos**, el cual almacena toda la información necesaria para el funcionamiento del sistema y en la parte cliente se sitúa el "*Gestor de esquemas*". El diccionario de datos contiene información sobre las bases de datos conectadas al servidor, los esquemas de las bases de datos, qué objetos de las bases de datos conectadas pueden ser accedidos por las aplicaciones clientes (los administradores de las bases de datos pueden decidir qué información comparten con el resto de usuarios), la definición y versiones anteriores de las clases clínicas, los enlaces de éstas con los esquemas de bases de datos, las relaciones semánticas entre clases clínicas, sinónimos, heurísticos para la optimización de las consultas y direcciones de red.

Una de las primeras y principales funciones del "*Gestor*" es la posibilidad de añadir nuevas BBDD al sistema (sus esquemas). La extracción y posterior importación del esquema de una base de datos se realiza vía JDBC. Es sabido que cada gestor de bases de datos habla en su propio lenguaje, normalmente dialectos del SQL estándar que implementa cada fabricante con sus variantes adaptadas a sus soluciones. No obstante actualmente podemos comunicarnos con estos gestores a través de un lenguaje común. La librería JDBC, nos facilita este lenguaje compartido. JDBC nos proporciona un conjunto de interfaces que crean un punto común en el que aplicaciones y gestores de BBDD pueden comunicarse mediante JDBC, pudiendo lanzar consultas a la base de datos con la que hemos conectado y recoger sus resultados. Entre las funcionalidades de JDBC esta también la de poder llegar a conocer como está diseñada la BBDD, o sea, sus tablas, campos, claves ajenas... información que recogeremos y guardaremos en la BBDDOO (diccionario de datos del sistema de integración), de esta manera el proceso de extracción se realiza una sola vez. El inconveniente de esto es que puede darse el caso que importe una BD en un momento dado y que posteriormente esa base de datos sea modificada en su ubicación original por las más diversas razones: se han añadido nuevas tablas, nuevos campos... Por esta razón el *gestor* permite "refrescar" el esquema de la base de datos conservando todo aquel trabajo realizado por los usuarios que es independiente del esquema original de la BD: tabla identificadora de paciente, disponibilidades, claves ajenas virtuales, descripciones etc

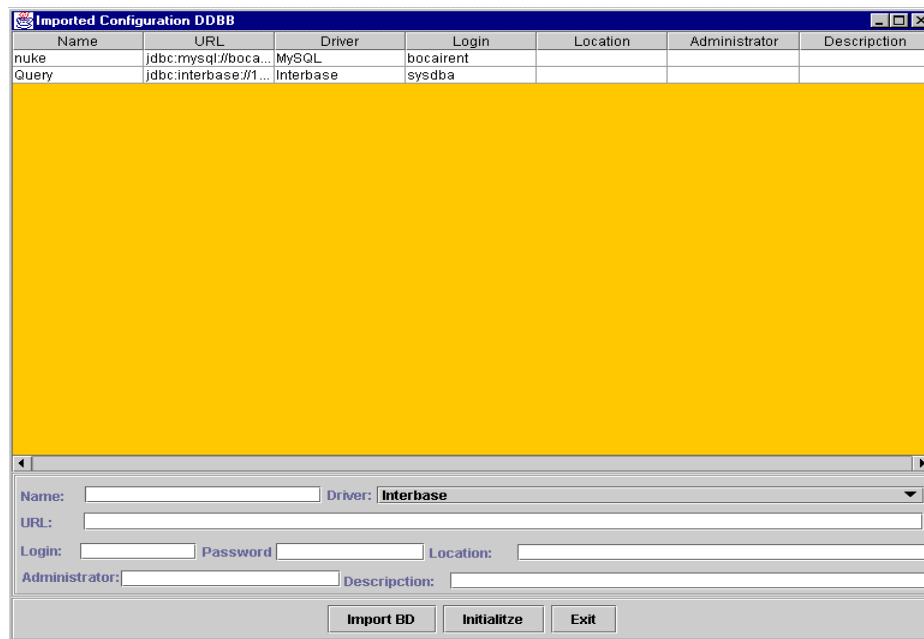


Figura 1. Pantalla desde la que importamos el esquema de una nueva BD.

Para poder importar el esquema de una BD necesitamos indicar los siguientes parámetros obligatorios: Nombre, Driver, URL, Login y Password. En la **figura 1** se muestra la pantalla de importación de esquemas, con el nombre indicamos eso mismo, como queremos nombrar a la BD a importar, en el árbol del *gestor de esquemas de BD visual* dónde aparecen todas las BBDD, de las cuales el sistema tiene información de su estructura, aparecerá bajo este nombre. En el driver indicaremos que driver JDBC utilizaremos para poder así comunicarse con la BD y extraer su esquema. La URL indica en que máquina esta la BD que queremos importar, en algunos casos será necesario indicar además del host o dirección IP el puerto por el cual el servidor de bases de datos nos dejará acceder a la BD en particular.

Una vez definidos estos parámetros obligatorios y según el caso alguno de los optativos ya podemos empezar el proceso de importación del esquema. La duración de esta operación variará dependiendo de muchos factores, del tamaño del esquema de la BD a importar sobre todo y después factores como el tipo de BD, saturación de la red, tipo driver JDBC... En todo caso generalmente la importación no suele llegar a un minuto salvo que la BD sea muy grande.

El *gestor de esquemas de bases de datos* en nuestro sistema se comunica directamente con el "Servidor de metainformación" el cual gestiona el diccionario de datos del sistema de integración. Este servidor gestiona tanto los esquemas de las bases de datos componente, la definición de los arquetipos y las relaciones entre ambas. En la figura 2 se muestra el esquema de comunicación entre todas estas componentes del sistema.

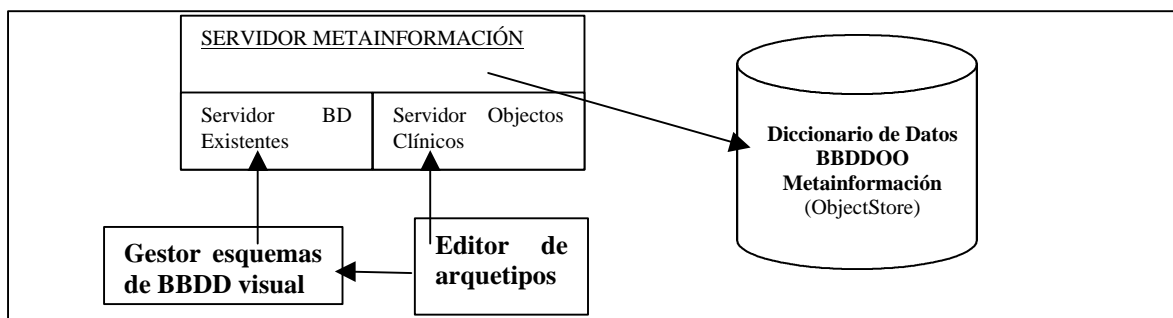


Figura 2. Esquema de comunicación entre las diferentes componentes que forman el sistema de integración y el gestor de esquemas de BBDD visual.

La **figura 3** muestra la apariencia del *gestor de esquemas visual*: En la parte izquierda aparece un árbol gráfico con todas las bases de datos importadas. En el ejemplo aparecen 2 bases de datos. Si desplegamos alguna de ellas aparecen las tablas pertenecientes a la base de datos en cuestión. Así por ejemplo, en la **figura 3** se muestran las tablas de la base de datos con nombre “ejemplo”.

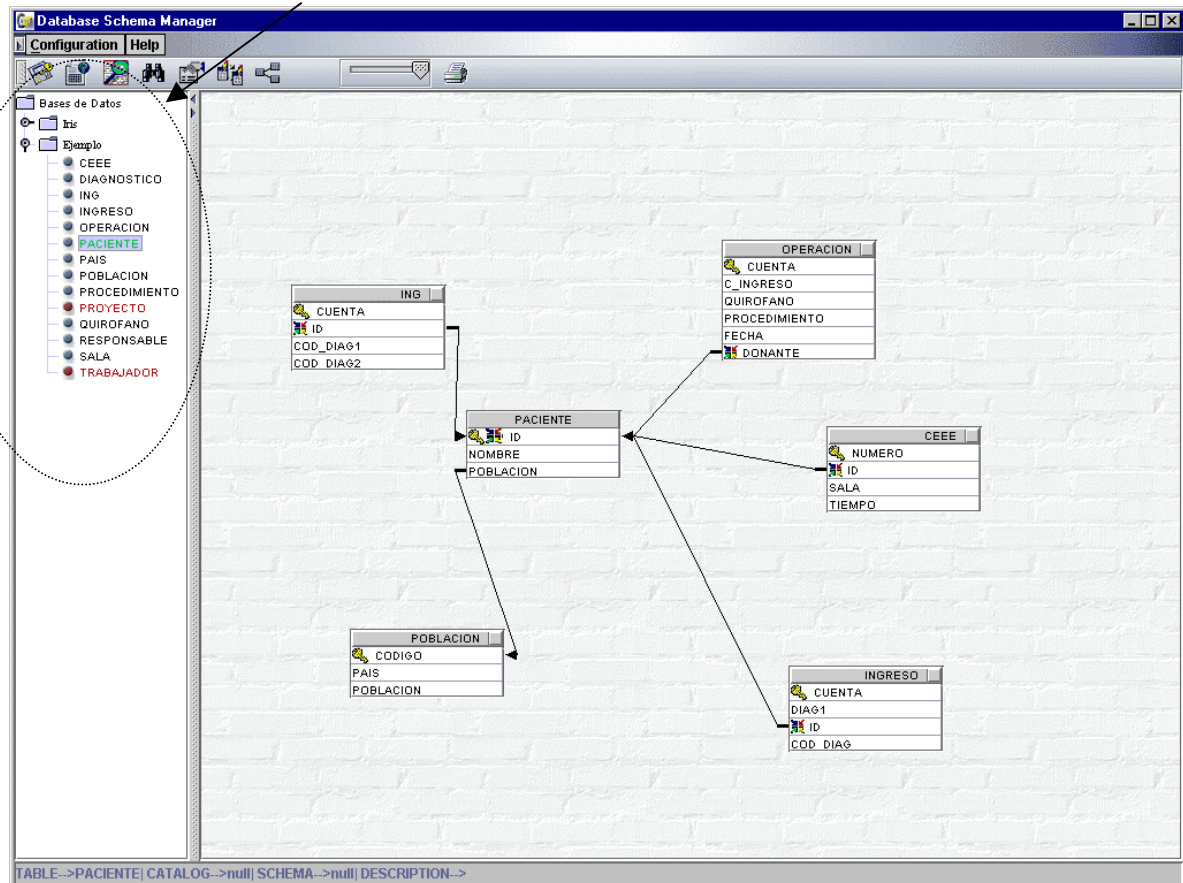


Figura 3. Pantalla Principal del “gestor de esquemas de BBDD visual”

Las tablas de las bases de datos pueden estar en alguno de estos cuatro estados:

- Tabla Identificadora de paciente.(representada con color verde, actualmente sólo 1 tabla por BD)
- Tabla no disponible. (representada de color ámbar)
- Tabla disponible. (representada de color azul)
- Tabla aislada. (representada de color rojo, este estado lo decide automáticamente el sistema)

Si desplegamos la base de datos (Ejemplo) y observamos sus tablas nos daremos cuenta que existe una tabla de color verde. Esta tabla se denomina *tabla identificadora de pacientes* que es definida por el usuario, esta será la que contenga como clave primaria el o los campos que identifiquen de forma unívoca a un paciente (campos identificadores de paciente). A partir de esta asignación se ha diseñado un mecanismo que detectará qué tablas de la BBDD contienen información relativa al paciente, y qué tablas deben considerarse aisladas (las de color rojo) desde el punto de vista del historial clínico, es decir, no contienen información clínica y por tanto no son relevantes. Cuando añadamos al sistema una nueva base de datos todas las tablas estarán aisladas en principio. Se necesita definir cual de todas las tablas de la nueva BD va a ser la *tabla identificadora de paciente*. Una vez definida una tabla el sistema es capaz de decidir que tablas están relacionadas directa o indirectamente con esta dejando el resto de tablas que no tienen relación en estado aislado (color rojo). Las que aparecen de color azul son tablas que están disponibles para el sistema, o sea, tablas que podremos

utilizar posteriormente con el *editor de arquetipos* para diseñar las plantillas de objetos clínicos. Las tablas que aparecen de color amarillento ocre son tablas no disponibles, pero que son accesibles (tienen relación directa o indirecta con la tabla maestra). El administrador puede alterar su estado a *no disponible* si no se desea que la información clínica contenida en ellas sea accesible a los usuarios finales. El cliente visual sólo muestra el estado de cada una de las tablas de las distintas BBDD que tiene incorporadas al sistema, el servidor de metainformación, es el que decide de manera automática o por peticiones desde el cliente el estado de estas, guardando en última instancia estos cambios en el diccionario de datos del sistema.

El gestor también nos permite visualizar gráficamente el conjunto de tablas que están relacionadas de manera directa con otra. Esta visualización se lleva a cabo de la siguiente manera, el cliente consulta al *Servidor de metainformación* (más concretamente al servidor de BBDD) que tablas están relacionadas mediante claves ajenas a una tabla X. El servidor lo consulta con la BBDDOO y reporta la información necesaria para realizar esta visualización. Como podemos comprobar todo tipo de acción que se quiera hacer desde el cliente previamente pasa como petición al Servidor quien en última instancia retorna y ejecuta las peticiones.

Como ya hemos comentado todas las tablas del sistema que aparecen aisladas (color rojo) son las que no tienen ninguna relación directa o indirecta con la tabla maestra. Puede ser por diversas circunstancias que alguna de estas tablas contenga información relevante para la futura creación de historias clínicas. ¿Qué pasa entonces? ¿No podemos utilizar esas tablas?. Para solucionar este handicap desde el cliente podemos definir relaciones (claves ajenas) entre dos tablas. De esta manera si esta clave ajena es sobre una tabla que esta directa o indirectamente relacionada con la *tabla maestra* la antigua tabla aislada pasará a formar parte del conjunto de tablas que tienen relación con la tabla maestra, su estado por tanto pasará a ser *disponible* o *no disponible* según se desee o convenga. El único estado de una tabla sobre el cual un usuario no puede interferir directamente es el estado *aislado*. Sobre el resto de los estados *tabla identificadora de pacientes*, *disponible* o *no disponible* si que se puede interactuar. No obstante el estado de *tabla identificadora de pacientes* maestra tiene una restricción, actualmente sólo puede ser una tabla de la base de datos (y debe ser la adecuada) aunque en un futuro podrán existir dos o más tablas maestras por BD.

Todo lo que hemos comentado hasta el momento ha girado en torno a las tablas. A nivel de bases de datos también podemos definir si esta estará o no disponible. También a nivel de campos de una tabla podremos realizar esta misma operación. Puede ocurrir que en una tabla disponible el administrador crea que ciertos campos no son necesarios para el diseño de historias clínicas por tanto puede cambiar su estado a *no disponible* por cuestiones de seguridad, confidencialidad u otras. Los campos además poseen la particularidad de poder ser definidos como campos demográficos. Un campo demográfico es aquel que contiene información relevante sobre un paciente tales como sexo, dirección, nombre, apellidos, fecha de nacimiento, número de seguridad social, número de historia clínica etc...

Conclusiones

El *gestor de esquemas de bases de datos visual* es una herramienta interesante ya no en el marco específico del proyecto en el cual esta enclavado. Si no como una aplicación útil para cualquier administrador que quiera de alguna manera importar esquemas de bases de datos y poder ver en un entorno amigable e integrado todos los esquemas así como buscar relaciones, redefinir claves ajenas, consultas SQL... En definitiva cualquier administrador de sistemas de bases de datos tiene a su disposición una herramienta capaz de integrar en una única vista información sobre las más diversas BBDD que utiliza una organización.

“El gestor de esquemas” es por tanto una aplicación digamos genérica pero con importantes matices para nuestro propósito específico. En nuestro caso concreto juega un papel imprescindible, es a partir del gestor desde dónde gestionamos qué información estará disponible y cual no, a la hora de diseñar los componentes clínicos de la arquitectura de la historia clínica electrónica además la visualización ayuda a entender los esquemas de las bases de datos componentes. También desde el gestor estableceremos los vínculos entre cada uno de los atributos o propiedades de un arquetipo de la arquitectura y un campo concreto de una BD.

Agradecimientos

Este proyecto está financiado por el ministerio de ciencia y tecnología, y por la comisión europea da través del proyecto FEDER-CICYT TIC2000-0706 y por Novasoft S.A.

BIBLIOGRAFIA

Luke Cassady-Dorion (1997), “Industrial Strength JAVA”. Ed. New Riders

Graham Hamilton, Rick Cattell, Maydene Fisher (1998). “JDBC Database Acces with Java”. Ed.- Sun Microsystems

Bouguettaya, A., Benatallah, B., Elmagarmid, A. (1998), *Interconnecting heterogeneous information systems*, Kluwer Academic Publishers.

Elmasri, R., Navathe, S. (2000), *Fundamentals of Database Systems*, tercera edición. Addison-Wesley.

Sheth, A.P., Larson, J.A., (1990). *Federated database systems for managing distributed, heterogeneous and autonomous databases*, ACM Computing Surveys, 22, pp. 183-235.