

Desarrollo de un aplicativo prototípico sobre arquitectura Corba para el almacenamiento y actualización de historias clínicas que se aproxime a Corbamed apoyándose en una de sus facilidades y con implementación en java

Henry Alberto
Diosa¹

Andrews Eduar do
Vargas Corredor²

William Pinilla
Ramos³

RESUMEN

Pensando en la necesidad de establecer un estándar en el proceso de identificación de pacientes en instituciones prestadoras de servicios de salud, de manera que dicho proceso sea más confiable y aplicable a entornos distribuidos, se desarrolló una aplicación para la gestión de historias clínicas electrónicas soportada en arquitectura CORBA (Common Object Request Broker Architecture), apoyándose en la facilidad de Identificación de Persona PIDS de CORBAMed. A la vez se realizó una propuesta en lenguaje de definición CORBA-IDL donde se especifican los tipos de datos y servicios estándar necesarios para garantizar que los registros clínicos ubicados en diferentes entidades hospitalarias con diversas plataformas tecnológicas puedan ser accedidos y manipulados; así mismo el uso de CORBA-IDL para la elaboración de la propuesta facilita la implementación de aplicativos computacionales en diversos lenguajes de implementación y para diversas plataformas.

El servicio de identificación de persona PIDS hace parte de las facilidades CORBAMed y es definido por el OMG mediante una especificación estándar. El producto desarrollado es un componente flexible y adaptable que utiliza los datos de identificación definidos por la legislación colombiana, pero puede ser configurado para utilizarse en otros contextos.

El proyecto se ha realizado con una previa investigación del estado del arte de tecnologías y proyectos orientados al área de la salud relacionados con la fuerza de tarea CORBAMed del OMG para el área de la salud.

Palabras clave: CORBA, CORBAMed, Historia Clínica Electrónica (HCE), Interface Definition Language (IDL), Personal Identification Service (PID).

ABSTRACT

Thinking about the need of establish a standard in the patient identification process in the health institutions, of manner that make this process more reliable and decentralized, an application for the manage of electronic clinic histories was developed supported on CORBA architecture, leaning in the CORBAMed facility PID Person Identification Service. At the same time, a propose in Interface Definition Language CORBA-IDL where made specifying the data type and standard services necessary to warranty that the clinic records located in different hospitable entities with diverse technological platforms may be acceded and manipulated; likewise the use of CORBA-IDL for the elaboration of the proposal, makes easier the implementation of computational applications in various languages of implementation and for various platforms.

PID Person Identification Service does part of the CORBAMed facilities and is defined by OMG through a standard specification. The developed product is a flexible and adaptable component that utilizes the data of identification circumscribed by Colombian legislation, but it can be configured to be used in another contexts.

The project has been made with a previous research about status of art of technologies and projects guided to the area of health, related with the task force CORBAMed of the OMG for the area of health.

1 Director grupo de investigación CORBA-OSM, Universidad Distrital Francisco José de Caldas.

2 Ingeniero de Sistemas, Universidad Distrital Francisco José de Caldas.

3 Ingeniero de Sistemas, Universidad Distrital Francisco José de Caldas.

La finalidad de la arquitectura CORBA es facilitar la creación de aplicaciones distribuidas, con una interoperabilidad entre componentes software independientemente del lenguaje en que están desarrollados, la plataforma hardware y el sistema operativo sobre el que se ejecutan.

Key Words: CORBA, CORBAMed, Electronic Clinic Histories (HCE), Interface Definition Language (IDL), Personal Identification Service (PID).

1. INTRODUCCIÓN

Las instituciones prestadoras de servicios de salud han venido manejando autónomamente la manera de identificación de los usuarios, almacenándolos y consultándolos de manera local, lo que conlleva a que estos identificadores carezcan de validez fuera de este ámbito, así cuando un paciente requiere ser atendido en una entidad de salud diferente a aquella en la cual tiene su historia clínica, el traslado de la misma puede ser imposible o demorarse demasiado tiempo.

Se requiere de un sistema que brinde la facilidad de ser ejecutado en las plataformas computacionales propias de cada entidad así como permitir a los equipos desarrolladores realizar su implementación en diversos lenguajes.

Lo ideal es pensar en una arquitectura que permita tener cada registro almacenado en la entidad a la cual pertenece el paciente, distribuyendo la carga de procesamiento y almacenamiento en cada una de las máquinas de la red pero integrándolos como un único sistema, brindando así transparencia de ubicación; también se debería crear un estándar para el manejo, almacenamiento y forma de acceso de la información de manera que se eviten inconvenientes a la hora del acceso a la misma.

De acuerdo a la problemática expuesta anteriormente, se ha desarrollado el proyecto consistente en una propuesta computacional que podría utilizarse como referencia para el proceso de estandarización del manejo de historias clínicas electrónicas en Colombia, apoyándose en las ventajas de una arquitectura distribuida como CORBA, haciendo uso del dominio del problema que posee la fuerza de tarea CORBAMed y utilizando una de sus facilidades.

2. MARCO CONCEPTUAL

2.1. OMG

OMG, es una entidad sin ánimo de lucro, creada en 1989 por varias compañías, con el fin de

promover la teoría y práctica de la tecnología de objetos en sistemas distribuidos de computación. En particular, está dirigida a reducir la complejidad, aminorar costos, y agilizar la introducción de nuevas aplicaciones de software. [1].

La misión del OMG es crear un mercado de software basado en componentes introduciendo las tecnologías de objetos y, a través de sus comités, establecer guías y especificaciones para proveer un marco común para el desarrollo de aplicaciones. Conforme a estas especificaciones, será posible desarrollar en ambientes heterogéneos a través de la gran variedad de productos hardware y software existente. Pero no solo publica estándares horizontales sino que tiene Fuerzas de Tarea que se especializan en áreas específicas haciendo un consenso entre clientes y desarrolladores y emitiendo especificaciones que son de utilidad para todos los implicados en estas áreas.

2.2. CORBA

CORBA es un *middleware* o marco de trabajo estándar y abierto de objetos distribuidos que permite a los componentes en la red interoperar en un ambiente común sin importar el lenguaje de desarrollo, sistema operativo, tipo de red, etc. En esta arquitectura, los métodos de un objeto remoto pueden ser invocados “transparentemente” en un ambiente distribuido y heterogéneo a través de un ORB (Object Request Broker). [2]

CORBA automatiza muchas tareas comunes de programación de redes tales como registro, localización y activación de objetos; manejo de errores y excepciones; codificación y decodificación de parámetros y protocolo de transmisión.

CORBA está constituido esencialmente de tres partes: un conjunto de interfaces de invocación, el ORB (Object Request Broker) y un conjunto de adaptadores de objetos (Objects Adapters). Las interfaces definen los servicios que prestan los objetos, el ORB se encarga de la localización e invocación de los métodos sobre los objetos y el Object Adapter es quien liga la implementación del objeto con el ORB.

Para que las interfaces de invocación y los adaptadores de objetos funcionen correctamente, se deben cumplir dos requisitos importantes. En primer lugar, las interfaces de los objetos deben describirse en un lenguaje común. En segundo lugar, todos los lenguajes en los que se quieran implementar los objetos deben proveer un mapeo entre los elementos propios del lenguaje de programación y el lenguaje común. La primera condición permite generalizar los mecanismos de pasaje de parámetros. La segunda permite relacionar llamadas de o a un lenguaje en particular con el lenguaje de especificación común. Este lenguaje común fue una parte esencial de CORBA desde sus orígenes y es conocido como el OMG IDL: Interface Definition Language.

CORBA ha logrado parte de su éxito a la clara separación entre la interfaz de los objetos y la implementación de los mismos. Las interfaces se definen utilizando el lenguaje IDL, cuya principal característica es su alto nivel de abstracción, lo que le separa de cualquier entorno de desarrollo específico. Para la implementación de los objetos se puede utilizar cualquier lenguaje de programación que proporcione enlaces con el lenguaje IDL. Para que un lenguaje de programación se pueda utilizar desde CORBA, debe tener definida la forma de enlazarse con IDL. [3]

Cuando el cliente y una implementación de objeto están distribuidos por una red, usan el protocolo GIOP/IOP (General Inter-ORB Protocol Internet Inter-ORB Protocol) suministrado por la arquitectura para lograr la comunicación.

2.3. CORBAMed

Es la fuerza de trabajo para el dominio de la salud creada por el OMG propone una serie de servicios que buscan solucionar problemas y a su vez mejorar la atención médica al paciente reduciendo costos.

CORBAMed define interfaces estandarizadas para muchos servicios de la salud orientados por objetos sobre las plataformas más comunes y disponibles en el dominio público. CORBAMed proporciona compatibilidad con una gran variedad de componentes software y acceso a un mercado más amplio de servicios especializados.

El objetivo de CORBAMed es mejorar el servicio de asistencia médica fomentando la interoperabilidad entre dispositivos de salud, expandiendo el conocimiento y uso de tecnologías CORBA por parte de las organizaciones de salud.

CORBAMed busca que sus facilidades sean extensibles a otras áreas de negocio, PIDS (Personal Identification Service) es una facilidad que no limita su uso al área de la salud, así mismo, la facilidad RAD (Resource Access Decision) de CORBAMed que constituye la facilidad enfocada a suplir la seguridad a la información clínica en el área de la salud, puede extender su aplicación a otras áreas brindando un servicio de control de acceso a los recursos. Estos dos son las facilidades CORBAMed más difundidas y en las que más adelantos se han realizado.

2.4. PIDS Person Identification Service

La especificación del Servicio de Identificación de Persona fue la primera especificación emitida por CORBAMed. Con ella se busca la "identificación de una persona" (paciente) con base en su ID (número mediante el cual se registró ante el sistema), o en una serie de parámetros de búsqueda (tales como nombre, dirección, fecha de nacimiento, entre otros). PIDS pretende establecer un único identificador por paciente o establecer una correlación de ID's para un mismo paciente, esto soluciona problemas como la duplicidad de registros médicos asociados al manejo autónomo de cada organización.

La especificación PIDS divide el alcance total de la "administración del identificador de persona" en componentes de servicio orientados a la responsabilidad:

- Identify Person: Dados algunos valores de búsqueda, encuentra los ID's y los perfiles para personas que coincidan con tales valores.
- ID Manager: "Administra" los identificadores dentro de una base o "dominio" de identificadores.
- Profile Access: Dado un ID, lee o escribe información del perfil, o accede a la información del estado del mismo.

- Correlation Manager: Busca coincidencias y correlaciona identificadores a lo largo de múltiples fuentes para su centralización.
- Sequential Access: Manipula grupos de identificadores, como por ejemplo los que se exportan e importan.
- Identity Access: Trata cada “identidad personal” como un recurso distinto de información.
- La información está disponible rápidamente lo cual es muy importante para la atención de urgencias.
- Permite presentar los datos de diversas formas por ejemplo de manera cronológica.
- Facilita la investigación médica

2.5.3. Desventajas

También tiene algunas desventajas:

- Existe riesgo de pérdida de integridad o disponibilidad debido a fallos técnicos.
- Requiere una mayor inversión inicial en hardware, software y formación del personal.
- La seguridad y confidencialidad de los datos son aspectos críticos especialmente cuando el sistema no se administrada adecuadamente.
- El personal puede presentar resistencia a las nuevas formas de introducción de datos y en general al cambio.

2.5. Historias Clínicas Electrónicas

2.5.1. Conceptos

La historia clínica podría definirse como el documento médico legal donde queda registrada toda la relación del personal sanitario con el paciente, todos los actos y actividades médico-sanitarias realizados con él y todos los datos relativos a su salud, que se elabora con la finalidad de facilitar su asistencia, desde su nacimiento hasta su muerte, y que puede ser utilizada por todos los centros sanitarios donde el paciente acuda [4].

Actualmente en la mayoría de centros de salud se utiliza la *historia médica automatizada* que utiliza el papel como soporte junto con un conjunto de sistemas informáticos departamentales aislados basados en arquitecturas tradicionales. Las *historias médicas computarizadas* consisten en el escaneo de los documentos basados en papel. Las *historias médicas electrónicas* comprenden un conjunto de sistemas de información departamentales pertenecientes a una única entidad hospitalaria que se encuentran integrados con un alto nivel de interoperabilidad.

Finalmente las *historias clínicas electrónicas* incluyen a las historias médicas electrónicas junto con enlaces multi-organización (regional, nacional, internacional). Por tanto requiere un sistema universal de identificación de pacientes y una infraestructura y tecnología para el intercambio de información.

2.5.2. Ventajas

Pueden destacarse:

- Toda la información sobre un paciente está disponible en varios lugares al mismo tiempo.

3. Proceso de desarrollo de un IDL

3.1. Perfil UML para CORBA

La Especificación del Perfil UML para CORBA fue diseñada con el fin de proveer un medio estándar para expresar la semántica de CORBA IDL usando notación UML y de esta manera soportar la expresión de esta semántica con herramientas UML. [5].

3.2. Proceso de Desarrollo de un IDL

3.2.1. Modelamiento de una aplicación Corba

El correcto modelamiento de una aplicación CORBA se debe basar en la especificación estándar emitida por el OMG “Perfil UML para CORBA”, se hace necesario disponer de una herramienta de modelamiento que soporte este perfil.

Lo primero que se hace es modelar cada uno de los tipos de datos CORBA mediante el uso de Clasificadores (generalmente Clases e Interfaces) incluyendo sus atributos y operaciones en caso que las tenga, a continuación y basados en la especificación se procede a estereotipar cada uno de estos clasificadores buscando una correspondencia directa entre el tipo de dato detallado en la especificación y el tipo de dato deseado.

Por último se procede a establecer las relaciones entre estos clasificadores teniendo en cuenta los criterios que proporciona la especificación para determinar si son relaciones de dependencia (como secuencias), relaciones de generalización (como en la herencia de interfaces), relaciones de agregación (como en el caso de algunas estructuras), además de relaciones de asociación, realización y composición según se requiera.

El resultado de este trabajo es un modelo estructural de datos tipo CORBA relacionados, que se realimenta y complementa progresivamente gracias a la elaboración de otros tipos de diagramas como pueden ser de secuencia, colaboración, actividades, etc.

3.2.2. Generación de archivo IDL

Basados en una herramienta en la cual se elabora y depura el modelo estructural de tipos de datos CORBA se procede a la generación mediante ingeniería directa del archivo IDL de tal modelo. La herramienta debe soportar el Perfil UML para CORBA para que la generación del archivo sea exitosa y efectivamente corresponda al modelo estructural base.

Se debe tener muy en claro la solución que se piensa modelar (dominio del problema) así como un conocimiento de los diferentes tipos de datos CORBA IDL y cómo deben ser éstos modelados para la generación exitosa del archivo IDL.

El resultado de este proceso es un archivo en lenguaje de definición (no de implementación) en el que se especifica las funcionalidades que debe brindar cualquier implementación que quiera estar conforme al mismo.

4. Instrumentación de PIDS de CORBAMED: SIDP-UD (Servicio de Identificación de persona – Universidad Distrital)

El componente SIDP-UD a implementar busca cumplir con la “clase de conformidad”⁴ SimplePIDS detallada en la especificación Person Identification Service PIDS version 1, publicada

⁴ Clases de conformidad: Puntos de referencia de programación para comprobar que el componente cumpla con el comportamiento esperado.

por el OMG en Abril de 2001. Los datos correspondientes a la historia clínica de un paciente que serán gestionados por defecto por el componente SIDP-UD son los especificados por el Ministerio de Salud en la resolución 1995 de 1999 y denominados Componente de Identificación de Usuario en el capítulo II, artículo 9.

4.1. Modelo Funcional

El modelo funcional resultante constó de 9 casos de uso, en donde se especifican las funcionalidades requeridas por el componente SIDP-UD, cada uno con su respectivo diagrama, descripción, escenario normal y cursos alternos, todo discriminado por actor.

Dichos actores son:

- Administrador SIDP-UD
- Aplicación de Historias Clínicas

En la Figura 1 se muestra un ejemplo del diagrama de caso de uso para el actor Aplicación Historias Clínicas.

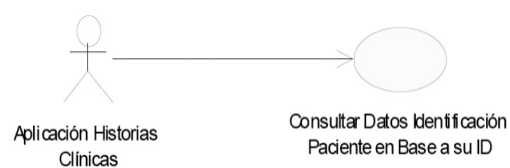


Figura 1. Caso de uso: Consultar datos de Identificación de Paciente en base a su ID.

4.2. Modelo Estructural

Como resultado de la realización del modelo estructural, se obtuvieron varias clases agrupadas funcionalmente en una serie de paquetes y subpaquetes. En este modelo encontramos clases y sus relaciones, dichos elementos pueden visualizarse por medio de diagramas, pueden ser de herencia, parte-todo y asociación.

Se tiene un modelo agrupado en tres paquetes básicos de implementación: paquete BD_UD que realizará la interacción con el repositorio de datos, paquete OMG IDL que contendrá los subpaquetes generados a partir de los archivos IDL especificados por el OMG y que constituyen los tipos de datos básicos que se utilizan para la implementación de SIDP-UD así como las

interfaces a implementar y por último, paquete SIDP_UDImpl que contendrá las clases de implementación de las interfaces PIDS, de conformidad al nivel SimplePIDS. En la Figura 2 se pueden observar estos paquetes, que además están detallados en el proyecto realizado. [6].

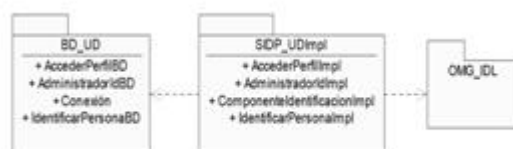


Figura 2. Paquetes utilizados en la implementación de SIDP-UD.

4.3. Modelo de comportamiento

Se describieron las realizaciones para los casos de uso, (modelar el comportamiento en términos de los objetos participantes), se realizaron diagramas de clases donde se visualizan las clases participantes en la realización junto con sus relaciones.

Posteriormente se realizan diagramas de secuencia que corresponden a la funcionalidad de cada caso de uso y en los que se puede observar las interacciones y asociaciones entre las diferentes clases. También se observa la secuencia de acciones que tienen lugar en caso de presentarse una excepción.

5. Propuesta computacional para la gestión de historias clínicas (GESTIHC-UD)

5.1. Modelo Funcional

El modelo funcional resultante constó de 44 casos de uso, en donde se especifican las funcionalidades requeridas por la propuesta computacional GESTIHC-UD, cada uno con su respectivo diagrama, descripción, escenario normal y cursos alternos, todo discriminado por actor.

Dichos actores son:

- Administrador
- Médico
- Digitador

En la Figura 3. se muestra un ejemplo del diagrama de caso de uso para el actor Médico.

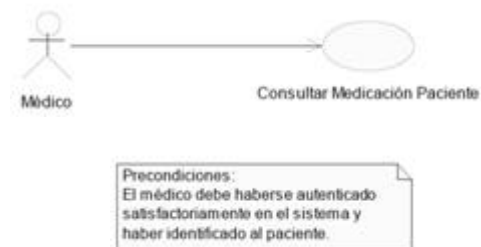


Figura 3. Caso de uso: Consultar Medicación de Paciente con sus respectivas precondiciones.

5.2. Propuesta especificación en IDL para la gestión de historias clínicas GESTIHC - IDL

5.2.1. Modelo Estructural

Se definieron ocho interfaces en las cuales se han agrupado los servicios facilitados por GESTIHC-IDL, tales interfaces son Interface Antecedentes, Interface Consultas Médicas, Interface Consulta Tratamientos, Interface Consulta Codificaciones, Interface Exámenes, Interface Hospitalizaciones, Interface Identidades y una interface principal denominada Interface GESTIHC. Además de lo anterior se han definido los tipos de datos que son utilizados por el sistema.

Se realizó la descripción del modulo CORBA GESTIHC-IDL, mediante un diagrama estructural del mismo, la descripción de sus interfaces así como la documentación de cada uno de los tipos de datos y excepciones definidas.

5.2.2. Modelo de comportamiento

Se describió la realización de cada caso de uso, utilizando los diagramas de secuencia y realizando el respectivo diagrama de clases, donde se visualizan las clases participantes en la realización junto con sus relaciones.

5.2.3. Ingeniería directa IDL

Una vez se obtuvo el modelo estructural para GESTIHC-IDL, se procedió a realizar el proceso de ingeniería directa, obteniendo de esta manera la especificación en lenguaje de definición

CORBA-IDL de la propuesta computacional para la gestión de historias clínicas electrónicas.

5.2.4. Aspectos de implementación

- Paquetización: En la elaboración de la propuesta en lenguaje de definición para la gestión de historias clínicas GESTIHCIDL se utilizó un único paquete (GESTIHCIDL) estereotipado como “CORBAModule” en el cual se encuentran todos los datos tipo CORBA así como las excepciones e interfaces (con sus servicios) propuestas que fueron modelados conjuntamente en un diagrama estructural.
- Utilización: El resultado del proceso de desarrollo realizado para lograr la propuesta computacional para la gestión de historias clínicas es un archivo de texto plano denominado GESTIHC IDL.idl.

5.3. Instrumentación de la aplicación para la gestión de historias clínicas basada en el IDL propuesto

GESTIHC-UD constituye la implementación realizada de la especificación GESTIHC-IDL, en el paquete GESTIHC-UD se detallan las clases utilizadas para llevar a cabo la implementación, así como el agrupamiento funcional realizado mediante la paquetización.

También se muestran las realizaciones para los casos de uso definidos en el modelo funcional mostrando las clases participantes en cada realización mediante un diagrama de clases y la forma en que éstas interactúan mediante diagramas de secuencia.

5.3.1. Modelo estructural

Se realizaron los diagramas tanto para la parte cliente como para la parte servidor, se identifican los siguientes paquetes:

- GESTIHC_IGU: En este paquete se agrupan todas las clases correspondientes a la Interfaz Gráfica de Usuario (IGU). Se recogerán los datos en tipo nativo de Java y se entregarán a la capa de clientes.
- GESTIHCCliente: En este paquete se encuentran todas las clases que actúan como clientes del servidor GESTIHC.

- GESTIHCImpl: En este paquete se agrupan las clases que implementan las interfaces definidas en la propuesta para el manejo de historias clínicas computacionales GESTIHC a través de GESTIHC-IDL. Se toman las peticiones con los datos provenientes de los clientes y recibidos a través del ORB. También se encarga de empaquetar los resultados en tipos de datos GESTIHC-IDL para ser retornados al cliente.
- GESTIHC_BD: En este paquete se han definido las clases que interactúan directamente con el repositorio de datos, se ha incluido una clase de interacción para cada una de las ocho interfaces definidas en el GESTIHC-IDL e implementadas por GESTIHC-UD.
- EXCEPCIONES : En este paquete se agrupan las clases de excepción propias de la implementación GESTIHCUD a ser lanzadas por las clases que interactúan con el repositorio de datos.
- GESTIHC_Utilidades: En este paquete se han reunido clases cuya funcionalidad es requerida por más de una clase, por lo cual requieren ser reutilizadas.

5.3.2. Modelo de comportamiento

Se describió la realización de cada caso de uso, utilizando los diagramas de secuencia y realizando el respectivo diagrama de clases, donde se visualizan las clases participantes en la realización junto con sus relaciones.

5.3.3. Modelo de persistencia

Se tomó como base un modelo relacional anterior [7] a partir del cual se elaboró un modelo mejorado, posteriormente se realizó un nuevo modelo relacional para motores de bases de datos multiplataforma como MySQL y Oracle.

Debido a que GESTIHC-UD es una implementación de GESTIHC-IDL, las clases persistentes equivalen a datos CORBA de tipo struct. Tomando como base el modelo estructural de GESTIHC-IDL (en el que se indican tanto clases persistentes como transitorias) se obtuvo un modelo entidad relación específico para la implementación GESTIHC-UD, este modelo consta de 58 entidades con sus respectivos atributos.

A partir del modelo entidad relación se obtuvo el modelo relacional que constituye el modelo físico de la base de datos en el que se especifican y detallan aspectos propios del motor de base de datos a utilizar.

5.3.4. Interfaz gráfica

La interfaz gráfica de usuario de GESTIHC -UD fue realizada tomando como base los gestores de distribución del paquete Swing: BorderLayout, GridLayout, FlowLayout, BoxLayout y GridBagLayout; así mismo en el desarrollo de estas interfaces se utilizó Java Swing.

En el paquete GESTIHC-IGU fueron incluidas 50 clases de interfaz gráfica de usuario, es decir la interacción de la implementación GESTIHC -UD, utiliza 50 ventanas para recoger y comunicar los datos al usuario, de igual forma en este paquete se encuentran todos los elementos gráficos que fueron utilizados en la implementación.

6. CONCLUSIONES

En la propuesta computacional en lenguaje de definición CORBA-IDL para la gestión de historias clínicas electrónicas, se especifican los datos y servicios estándar que deben ser proporcionados, tomando como base los requerimientos y necesidades que existe en el ámbito colombiano en esta área, con lo que podría servir de referencia para futuros proyectos, o para un posible proceso de estandarización de la Historia Clínica Electrónica.

Debido a que la propuesta desarrollada se encuentra definida mediante lenguaje IDL facilitará que las organizaciones interesadas en ella puedan realizar su instrumentación utilizando el lenguaje de implementación y la plataforma computacional que consideren más conveniente o necesaria sin que ello impida la comunicación e intercambio de información con otras entidades.

Se realizó el desarrollo e instrumentación de una aplicación prototípica CORBA, que implementa la propuesta computacional para la gestión de HCE, esta solución cumple con los requisitos médicos y legales y puede llegar a ser utilizada por grupos de entidades prestadoras de salud cuyas historias clínicas pretendan ser unificadas, como es el caso de la Secretaría Distrital de Salud de Bogotá que busca compartir

la información de los registros clínicos de los pacientes que pertenecen a la red mediante el programa de Cooperación Científica.⁵

La aplicación descrita hace uso de la facilidad de identificación de persona PIDS de CORBAMed, con la cual es posible realizar la identificación de un paciente con base en su ID o sus rasgos. Debido a que tal facilidad esta estandarizada por el OMG, sus tipos de datos y servicios son comunes para todas las instrumentaciones, pudiendo de esta manera utilizar cualquier componente PIDS disponible cuyo nivel de conformidad sea como mínimo SimplePIDS.

Los desarrollos descritos fueron realizados mediante el uso de herramientas libres y versiones de prueba, lo cual permitirá a cualquier persona o entidad interesada revisarlos con fines académicos. De igual manera para el proceso de desarrollo se aplicaron los lineamientos definidos por el Proceso de Desarrollo Unificado RUP con el fin de asegurar la calidad de los productos.

REFERENCIAS

- [1] Kate Keahey, A Brief Tutorial on CORBA, disponible en <http://www.cs.indiana.edu/~kksiazek/tuto.html>
- [2] Yamile Ramirez Herrera, Componentes, disponible en <http://www.monografias.com/trabajos16/componentes/componentes.shtml>
- [3] Alvaro del Castillo, Curso práctico de Corba en GNU/Linux, disponible en http://www.programacion.com/tutorial/acscorba/3/#corba_quees
- [4] Dolores Giménez Pérez, Especialista Medicina Legal y Forense, La historia clínica, aspectos médicos y legales, disponible en <http://www.ub.es/fildt/historia.htm>.
- [5] Información sobre el perfil UML para CORBA del OMG disponible en <http://www.omg.org/technology/docurnents/fornal/profilecorba.htm>

⁵ Cooperación científica: Relación de confianza que se establece entre varias entidades de salud mediante la cual se permite compartir cierta información clínica privada con el fin de mejorar la calidad en la atención al usuario y colaborar con la investigación científica.

[6] Vargas Corredor, Andrews Eduardo; Pinilla Ramos, William. "Revisión documental y desarrollo de un aplicativo prototípico sobre arquitectura Corba para el almacenamiento y actualización de historias clínicas que se aproxime a Corbamed apoyándose en una de sus facilidades y con implementación en java". Proyecto de grado. Director: Diosa, Henry Alberto. Universidad Distrital Francisco José de Caldas. Bogotá. 2003

[7] Tolosa Peña, Diana Lucia; Delgado Miguel. Aplicación para el almacenamiento y administración de historias clínicas sobre plataformas PC y agendas digitales personales con sistema operativo Windows CE. Trabajo de grado. Universidad Distrital Francisco José de Caldas. Proyecto Curricular de Ingeniería de Sistemas. Bogotá, 2001, 15 p.

Henry Alberto Diosa

Director grupo de investigación CORBA-OSM, Universidad Distrital Francisco José de Caldas.

Andrews Eduardo Vargas Corredor

Ingeniero de Sistemas, Universidad Distrital Francisco José de Caldas.

William Pinilla Ramos

Ingeniero de Sistemas, Universidad Distrital Francisco José de Caldas.