Una arquitectura de software para la gestión de historias clínicas electrónicas soportada en tecnologías web

Henry Alberto Diosa[1]

Carlos Andrés Isaza Peña[2]

Teresa Buenhombre González[3]

RESUMEN

Entre las actividades que viene realizando, el Grupo CORBA-OSM ha asumido el desarrollo de algunos proyectos de grado que exploren el potencial de las tecnologías Web. El sistema de "Gestión de Historias Clínicas Electrónicas Soportado en Tecnologías WEB" es una aplicación que busca plantear un posible soporte al tratamiento y gestión de las historias clínicas dentro de una Institución Prestadora de Servicios de Salud (IPS), específicamente en el marco de la consulta médica general o consulta externa, mediante la utilización de herramientas de software que giran alrededor de la WEB, tales como: Java, JSP (Java Server Pages), XML (Extensible Markup Language) y sus tecnologías relacionadas.

Palabras clave: XML(eXtensible Markup Language), JSP(Java Server Pages), DOM(Document Object Model),HCE(Historia Clínica Electrónica), UML(Unified Modeling Language), Contenedor de Servlets, XML-Schema, Arquitectura.

ABSTRACT

Among the activities that it comes carrying out, the CORBA-OSM group has assumed the development of some degree projects that explore the potential of Web technologies. The supported management system of Electronic Clinical Histories in Web Technologies is an application that looks for to outline a possible support to the treatment and administration of the clinical histories inside a "Institución Prestadora de Servicios de Salud (IPS)", specifically in the mark of the general medical consultation or external consultation, by means of the use of software tools that rotate around the WEB, such as: Java, JSP (Java Server Pages), XML (Extensible Markup Language) and their related technologies.

Key Words: XML(eXtensible Markup Language), JSP(Java Server Pages), DOM(Document Object Model),ECH(Electronic Clinical Histories), UML(Unified Modeling Language), Servlet Container, XML-Schema, Software Architecture

I. INTRODUCCIÓN

El Grupo CORBA-OSM viene asumiendo el estudio e investigación de temas relacionados con tecnologías "middleware" con énfasis en arquitecturas soportadas en CORBA(Common Object Request Broker Architecture) y OSM(Objetos de Software Móviles). Por lo promisorio de las tecnologías Web como bus comunicante de aplicaciones distribuidas se han abordado otros proyectos alrededor de este tema como el que nos ocupa en este artículo. Así mismo, una de las decisiones al iniciar los trabajos del grupo fue elegir un eje temático-problémico para aplicar las tecnologías estudiadas y efectuar el ejercicio académico; el eje problémico elegido fue la sistematización de la historia clínica con formato de medicina general. Por esta razón, se realizó un primer proyecto que permitió el estudio normativo nacional e internacional de la manipulación de la información que contiene la historia clínica y se concretó en el desarrollo de una aplicación tecnológica que permite transportar dicha historia clínica en una Agenda Digital Personal y gestionarla desde un computador personal usando para el modelamiento UML(Unified Modeling Language), para la implementación de la interfaz de usuario y lógica de aplicación el lenguaje de programación JAVA y un repositorio de datos en ACCESS[1]. Este proyecto aportó un modelo estático de historia clínica que fue reutilizado en el proyecto que a continuación se describe, obviamente se hicieron mejoras al mismo como parte del aspecto evolutivo del modelo y de las nuevas funcionalidades soportadas.

II. LA ARQUITECTURA

El sistema de administración de historias clínicas presenta una arquitectura de tipo cliente robusto en la que se gestiona el servicio de aplicación a través de un contenedor de Java Server Pages(JSP's) denominado Apache Tomcat (perteneciente al proyecto Apache)[2] con una correspondencia directa con Java, que utilizando tecnologías de procesamiento de documentos XML permite la manipulación de la historia clínica a los clientes que solicitan el servicio con sus debidas restricciones de acuerdo al perfil de usuario. La Figura 1 ilustra una visión arquitectónica que hace énfasis en los protocolos de interacción entre el cliente y el servidor junto con los pasos de procesamiento que se dan en cada nodo.

El procesamiento de peticiones de los clientes al servidor se realiza a través del protocolo HTTP (Hypertext Transfer Protocol) y es llevado a cabo por medio de páginas de servidor (JSP), cuyo único

El sistema de administración de historias clínicas presenta una arquitectura de tipo cliente robusto en la que se gestiona el servicio de aplicación a través de un contenedor de Java Server Pages.

[1] Director grupo de investigación CORBA-OSM, Universidad Distrital Francisco José de Caldas.
[2] Morris Technologies.
[3] Proyecto Taritel.

Artículo recibido en Septiembre 10 de 2002
Aceptado el 13 de Noviembre 2002

El uso de tecnologías que permiten una arquitectura abierta hace promisoria la extensión de la misma a otras áreas relacionadas con la gestión de historias clínicas electrónicas.

trabajo es el de gestionar los datos ingresados para direccionarlos a la base de datos a través de clases Java. El direccionamiento de los datos y su correspondiente validación se realiza en el lado del cliente desde páginas HTML (Hyper- text Markup Language) y la ejecución de los métodos de procesamiento la hace Java. Java permite manipular los datos recibidos e ingresarlos en la base de datos del sistema a través de un puente ODBC - JDBC.

La consulta de la historia clínica se realiza mediante el mismo procedimiento (HTML-JSP-Java-SQL y SQL-Java-JSP-HTML) pero a diferencia del ingreso de datos para actualización, se deben incorporar al proceso las tecnologías de manipulación de XML que permiten separar la vista lógica del negocio, de la forma de presentación y contenido; cuestión que favorece la velocidad del desarrollo de aplicaciones ya que el analista se preocupa por la lógica de la aplicación y la tarea de visualización se la deja completamente al diseñador. Así mismo se genera una capa intermedia, independiente del manejador de base de datos subyacente, para propósitos de validación estructural de contenidos que facilita la estandarización del documento electrónico y además con la capacidad de ser repositorio de datos en si misma, esto es bastante útil porque se vienen desarrollando investigaciones de bases de datos Web soportadas en XML con el objeto de extraer datos semánticamente útiles, que en este caso aplicarían a las ontologías propias de la medicina[3].

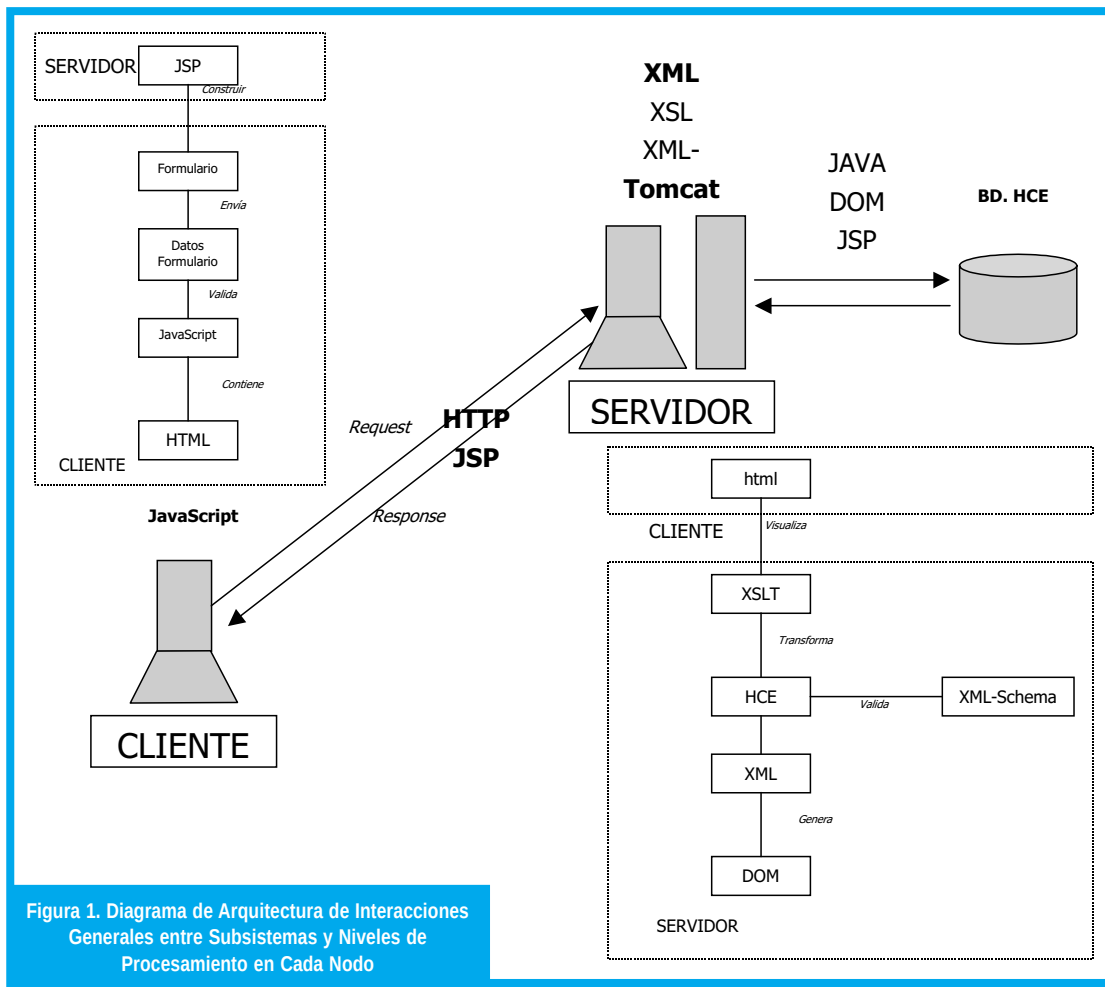


Figura 1. Diagrama de Arquitectura de Interacciones Generales entre Subsistemas y Niveles de Procesamiento en Cada Nodo

La arquitectura de red (ver figura 2) pone en evidencia la posibilidad de integrar fácilmente los potenciales usuarios del sistema(consultorios, pacientes, laboratorios, enfermería) usando la plataforma de Internet. Obviamente esta arquitectura puede transportar volúmenes de datos de diferentes órdenes de magnitud de acuerdo a la capacidad neta de transmisión de los enlaces; sin embargo, esta versión de la aplicación no es muy exigente frente al ancho de banda requerido porque sólo se manejan datos de tipo texto. Lo anterior no quiere decir que la arquitectura del software planteada no se pueda extender a manejar la transmisión de datos de imágenes diagnósticas y otro tipo de datos exigentes en ancho de banda neto de transmisión. El uso de tecnologías que permiten una arquitectura abierta hace promisoria la extensión de la misma a otras áreas relacionadas con la gestión de historias clínicas electrónicas.

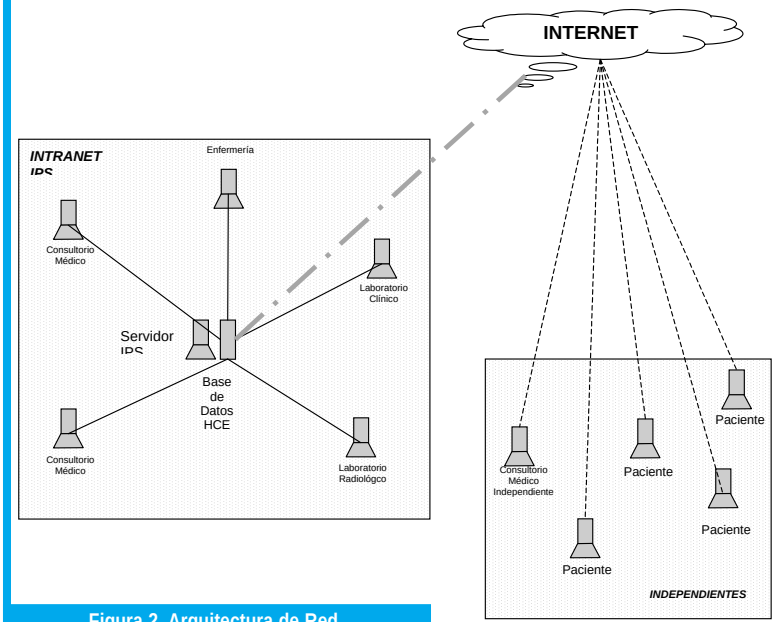


Figura 2. Arquitectura de Red

III. EL MODELAMIENTO

Para el desarrollo de esta aplicación se usó el modelo de proceso unificado con algunas simplificaciones para favorecer la viabilidad del proyecto abordado con la cantidad de personal involucrado.

El modelamiento de la aplicación se desarrolló con base en el Lenguaje de Modelamiento Unificado (Unified Modeling Language - UML) que permitió conceptualizar el sistema analizado en modelos concretos independientes de la plataforma de desarrollo. Se elaboraron tres modelos base que conformaron la parte sustancial del diseño total del software, éstos modelos son:

- Modelo funcional - diagramas de casos de uso.
- Modelo estático - diagramas de clases y asociaciones.
- Modelo dinámico - diagramas de secuencia.

El desarrollo del modelo funcional se realizó con base en una investigación y análisis preliminar de los objetos y del proceso de negocio consulta médica general.

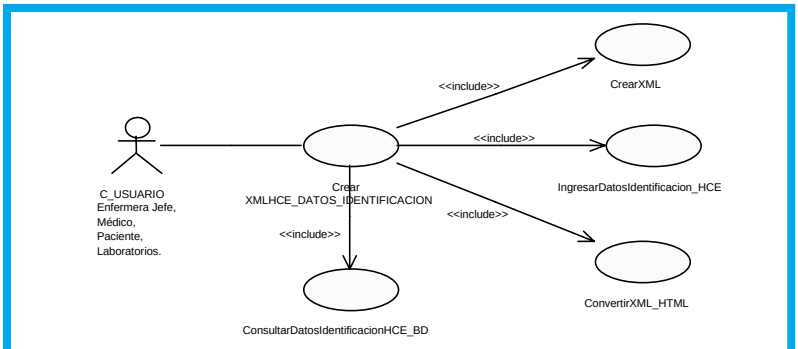


Figura 3. Diagrama de caso de uso Crear XML HCE Datos de Identificación.

El modelo funcional generado dio como resultado veintiocho(28) casos de uso, que conceptualizaron el modelo de operaciones de usuario(Los usuarios determinados dentro del proceso de investigación fueron: Médico general, Enfermera jefe, Paciente, Administrador paraclínico y Administrador del sistema) [4] que plasma las actividades que éstos llevan a cabo frente al sistema. Por ejemplo, uno de los casos de uso más representativos fue el de la creación del documento XML HCE Datos De Identificación que es llevado a cabo por los diferentes usuarios que interactúan directamente con el documento clínico. La Figura 3 muestra el diagrama correspondiente con el caso de uso mostrando las diferentes operaciones que se desencadenan a partir de la solicitud del documento.

El modelo estático del sistema representa la estructura de contenidos de la historia clínica electrónica mediante diagramas de clases, especificando sus atributos y relaciones entre ellas. Este modelo estructural alimenta la definición de validadores estructurales de los contenidos de una historia clínica tales como los XML-Schema definidos para este caso (Para mayor claridad ver [5]). Como un ejemplo, la Figura 4 visualiza la parte del diagrama de clases del sistema relacionado directamente con el caso de uso en cuestión. La clase CHceId constituye los datos de identificación de la historia clínica y del paciente, su estructura jerárquica visualiza los datos del paciente y de la Entidad Prestadora de Servicios de Salud a la cual se encuentra adscrito el paciente. Dentro de sus operaciones se pueden identificar: conectar con base de datos, insertar nuevo registro de HCE en base de datos y obtener datos de identificación de HCE para la construcción del árbol DOM y su correspondiente transformación en HTML.

El modelo dinámico está compuesto por diferentes diagramas de secuencia, los cuales corresponden en funcionalidad con cada uno de los diagramas de casos de uso -establecidos en el modelo funcional- y en lógica de aplicación -modelo estático- con las interacciones y asociaciones entre las diferentes clases del diagrama de clases parte todo del sistema. Este modelo, además establece las asociaciones entre el modelo funcional y el modelo estructural de la aplicación. Como un ejemplo, la Figura 5 muestra el diagrama de secuencia para el caso de uso de Crear XML HCE Datos de Identificación, en este diagrama se visualizan los diferentes estereotipos que corresponden a las notaciones convencionales de UML y a las notaciones extendidas de UML para aplicaciones WEB propuestas por Jim Conallen [6].

La última etapa asumida fue la construcción o implementación del diseño aprobado como línea base de trabajo.

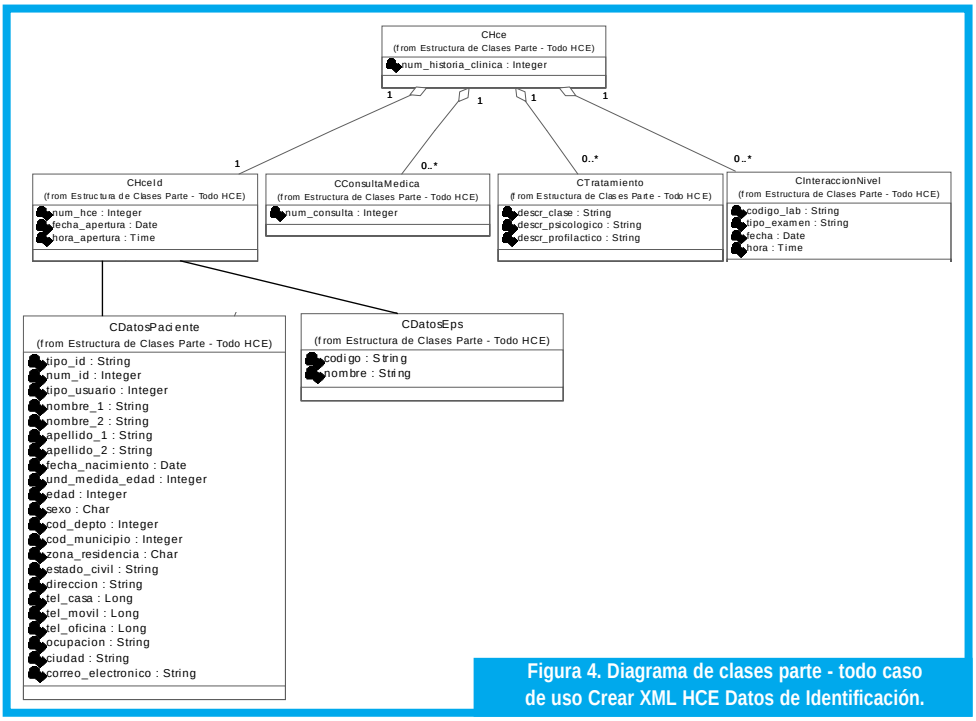


Figura 4. Diagrama de clases parte - todo caso de uso Crear XML HCE Datos de Identificación.

IV. LA IMPLEMENTACIÓN

La construcción de la aplicación conllevó la construcción de diferentes interfaces que permiten la interacción directa entre el usuario y la aplicación. Estas interfaces son las encargadas de enviar las peticiones del cliente, mediante HTTP, al servidor de aplicación que las recibe a través de páginas JSP que contienen objetos de las clases incorporadas en el modelo estático y que recibe las opciones ingresadas por parte del usuario para la consulta de las HCE´s en el sistema.

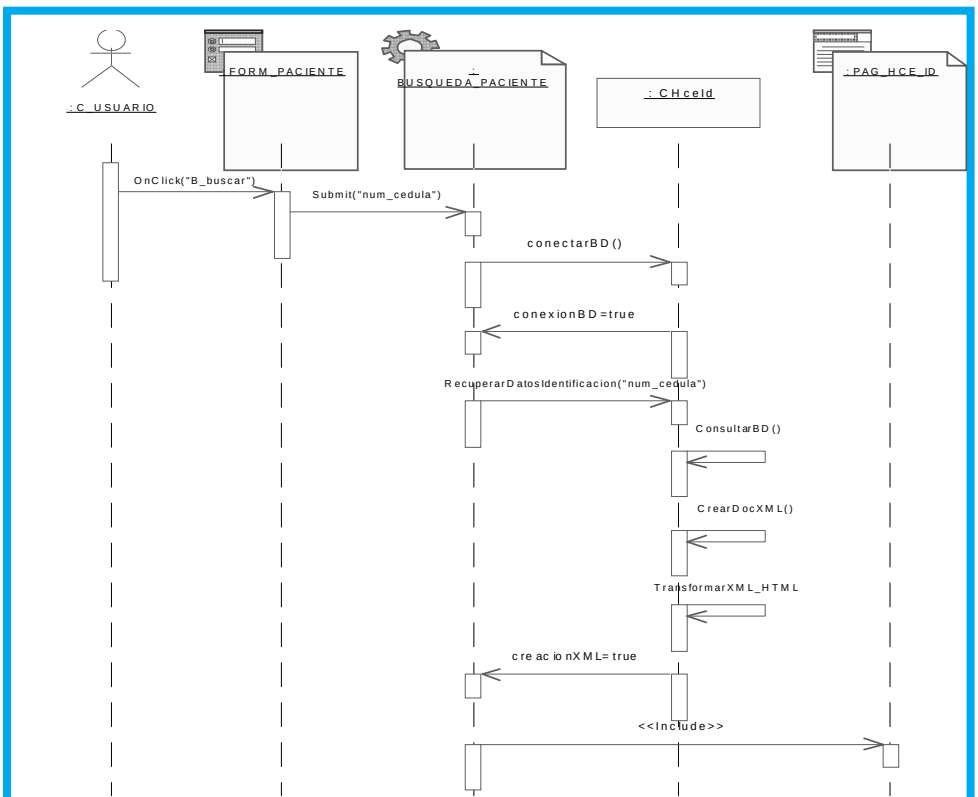


Figura 5. Diagrama de secuencia para el caso de uso Crear XML HCE Datos de Identificación.

Dentro de las herramientas utilizadas para la aplicación de gestión de historias clínicas electrónicas se utilizó como procesador, analizador y transformador de documentos XML [5] la herramienta JAXP 1.2 (Java API for XML Processing) [7]. Esta interfaz permite manipular los documentos XML desde Java mediante el analizador DOM (Document Object Model) que permite construir una representación en memoria de objetos con los datos contenidos en el documento. Además soporta el estándar XSLT (XML Stylesheet Language Transformations) permitiendo así el control sobre la presentación de los datos y permitiendo además convertir los datos del formato XML a otros formatos tales como HTML, PDF (Portable Document Format), WML (Wireless Markup Language), VML (Vector Markup Language), entre otros.

JAXP permite trabajar con espacios de nombres que de no utilizarlos se crearían conflictos de nombrado en los documentos XML, esto se realiza a través del estándar XML Schema soportado por esta API.

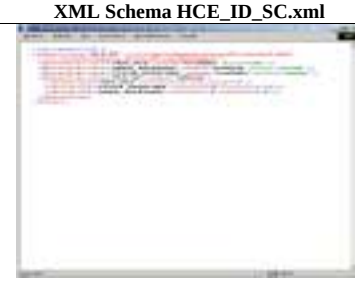
La estructura cliente/servidor de la aplicación para la consulta de la HCE requiere de una secuencia de procesos desencadenados por el usuario que van desde la recepción de las opciones de consulta en una página HTML, pasando por la recepción y validación de los datos en una página JSP, transmitiendo las opciones a una clase Java que consulta la base de datos con SQL que retorna los datos resultantes de la base de datos a la clase Java que construye un documento XML (a través de la interfaz DOM de JAXP) . La Figura 6 muestra el método de creación del árbol con JAXP recuperando iterativamente los datos de la base de datos.

```
//Crea y configura una fábrica constructora de documentos:
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setValidating(true); //Con validación.
dbf.setNamespaceAware(true); //Con espacios de nombres (XML Schema).

//Crea un constructor de documentos XML:
DocumentBuilder db = dbf.newDocumentBuilder();

//Carga un encabezado y estructura de un documento XML existente y obtiene su Nodo Raiz:
Document hceIdXML = db.parse("C:\\Archivos de
            programa\\Tomcat4.0\\webapps\\sghce\\jsp\\Administrador\\HCE\\HCE_ID\\HCE_ID.xml");
Element Raiz_hceId = hceIdXML.getDocumentElement();

//Conecta con la base de datos y consulta registro de HCE:
Connection consultarHceID = DriverManager.getConnection("jdbc:odbc:BD_SGHCE");
Statement declaracion = consultarHceID.createStatement();
String sentenciaSQL = "SELECT HCE_ID.NUM_HISTORIA, HCE_ID.FECHA_APERTURA,
            HCE_ID.HORA_APERTURA, DATOS_PACIENTE.SEXO,...(demás datos...)
            FROM HCE_ID, DATOS_PACIENTE, DATOS_EPS, USUARIO
            WHERE HCE_ID.NUM_HISTORIA=DATOS_EPS.NUM_HISTORIA AND,
            DATOS_PACIENTE.NUM_HISTORIA = HCE_ID.NUM_HISTORIA
            0AND USUARIO.CEDULA = '"+num_hce+"' AND USUARIO.PERFIL = 'Paciente' AND
            HCE_ID.NUM_HISTORIA = '"+num_hce+"'";
ResultSet resultado_consulta = declaracion.executeQuery( sentenciaSQL );
while ( resultado_consulta.next() )
{
   //Recupera cada uno de los resgistro de la base de datos:
   String nhc = new String ( resultado_consulta.getString("NUM_HISTORIA") );
   String fec = new String ( resultado_consulta.getString("FECHA_APERTURA") );
   String hor = new String ( resultado_consulta.getString("HORA_APERTURA") );
   .
   .
   .
   //Crea los elementos necesarios (según el esquema asociado) dentro del documento XML cargado:
   Element E_hce = hceIdXML.createElement(«HCE_ID:NUM_HCE»);
   Element E_fec = hceIdXML.createElement("HCE_ID:FECHA_APERTURA");
    Element E_hor = hceIdXML.createElement("HCE_ID:HORA_APERTURA");
   .
   .
   .
   //Adiciona los atributos a cada uno de los nodos respetando el esquema XML:
   E_datosPaciente.setAttribute("TIPO_USUARIO", tip);
   E_identificacion.setAttribute("TIPO_ID", tid);
   E_datosIdentificacion.setAttribute("SEXO", sex);
   E_datosIdentificacion.setAttribute("ESTADO_CIVIL", est);
   E_eda.setAttribute("UND_MEDIDA", ued);
   E_datosResidencia.setAttribute("ZONA_RESIDENCIA", zre);

   //Asigna la estructura de árbol DOM:
   Node N_hce = E_hceid.appendChild(E_hce);
   Node N_fec = E_hceid.appendChild(E_fec);
   Node N_hor = E_hceid.appendChild(E_hor);
   Node N_datosIdentificacion = E_datosPaciente.appendChild(E_datosIdentificacion);
   Node N_identificacion = E_datosIdentificacion.appendChild(E_identificacion);
   .
   .
   .
   //Crea los nodos de tipo texto  requeridos según el esquema XML:
   Text T_hce = hceIdXML.createTextNode("HCE_ID:NUM_HCE");
   Text T_fec = hceIdXML.createTextNode("HCE_ID:FECHA_APERTURA");
   .
   .
   .
   //Asigna los valores de los nodos de texto con los valores recuperados de la base de datos:
   T_hce.setNodeValue(nhc);
   T_fec.setNodeValue(fec);
   T_hor.setNodeValue(hor);
   T_n1.setNodeValue(n1+" ");
   .
   .
   .
   //Asigna los nodos de texto al árbol DOM ensamblado:
   N_hce.appendChild(T_hce);
   N_fec.appendChild(T_fec);
   .
   .
   .
}
resultado_consulta.close();
declaracion.close();
consultarHceID.close();

//Crea y configura una fábrica de transformacion de documentos:
TransformerFactory tf = TransformerFactory.newInstance();
Transformer t = tf.newTransformer();

//Carga en un flujo XML el árbol cargado en memoria:
DOMSource s = new DOMSource (hceIdXML);

 //Crea el nuevo archivo XML que se generará de DOM creado:
File hceXML = hceXML = new File ("C:\\Archivos de
            programa\\Tomcat4.0\\webapps\\sghce\\jsp\\Administrador\\HCE\\HCE_ID\\HCE_IDbd.xml");
FileOutputStream os = new FileOutputStream(hceXML);
StreamResult r = new StreamResult(os);

//Transforma el árbol DOM en un archivo XML en disco:
t.transform(s,r);

//Crea y configura una fábrica de transformacion de documentos:
TransformerFactory tf2 = TransformerFactory.newInstance();
String hceIdXSL = "";

//Carga la ruta relativa a la hoja de estilo prediseñada dependiendo del perfil de acceso del usuario al sistema:
if ( opcion == 1 )
    hceIdXSL = "C:\\Archivos de
            programa\\Tomcat4.0\\webapps\\sghce\\jsp\\Administrador\\HCE\\HCE_ID\\HCEIdentificacion.xsl";
else
    hceIdXSL = "C:\\Archivos de
            programa\\Tomcat4.0\\webapps\\sghce\\jsp\\Administrador\\HCE\\HCE_ID\\HCE_ID_EJ.xsl";

 //Carga la ruta del archivo XML generado anteriormente de disco duro:
String ShceIdXML = "C:\\Archivos de
            programa\\Tomcat4.0\\webapps\\sghce\\jsp\\Administrador\\HCE\\HCE_ID\\HCE_IDbd.xml";

//Crea el nuevo archivo HTML en el que se depositará la transformación:
File hceIdHTML = new File ("C:\\Archivos de
            programa\\Tomcat4.0\\webapps\\sghce\\jsp\\Administrador\\HCE\\HCE_ID\\HCE_IDbd.html");
FileOutputStream os2 = new FileOutputStream ( hceIdHTML );
Transformer t2 = tf2.newTransformer( new StreamSource ( hceIdXSL ) );

//Transforma el archivo recurso XML a traves de la XSL en un archivo HTML:
t2.transform ( new StreamSource ( ShceIdXML ), new StreamResult( os2 ) );
```

Figura 6. Creación del árbol DOM a través de la interfaz DOM de JAXP y Código de Transformación Automática en XML y luego a HTML.

El documento es validado por la fábrica de documentos con tres esquemas XML diferentes (Figura 7).

XML Schema HCE_ID_SC.xml



DATOS_PACIENTE_SC.xml

DATOS_EPS_SC.xml

Documento HCE_ID.xml

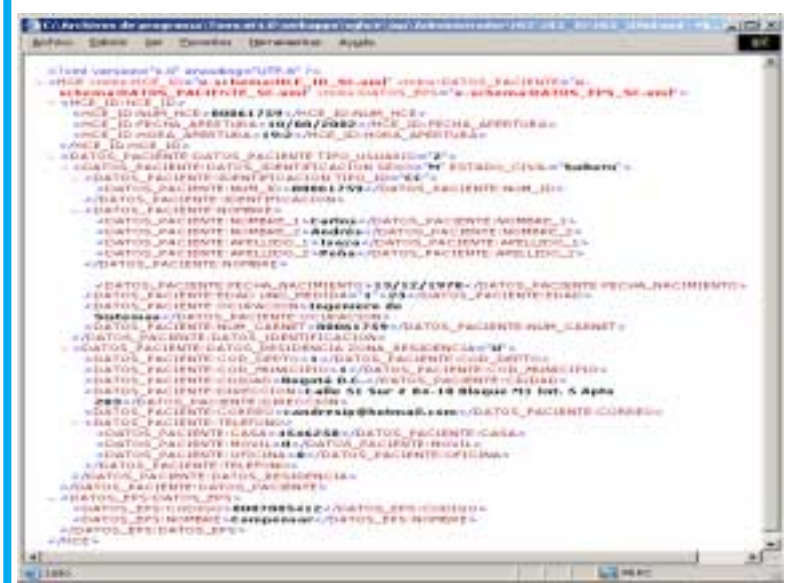


Figura 7. Documentos XML Schema de validación del documento HCE_ID.xml.

Y lo transforma en un nuevo documento en formato HTML, pasándole como atributo la ruta a una hoja de estilo (XSL) que apoyará la transformación (Figuras 8 y 9).

V. CONCLUSIONES

1) Las tecnologías Web presentan un soporte a la viabilidad del tratamiento de las historias clínicas electrónicas en Internet para el ambiente de una IPS dentro de la consulta externa de un paciente afiliado alsistema de salud. El proyecto de gestión de historias clínicas electrónicas presenta además una propuesta de modelo de historia clínica electrónica para su correspondiente manipulación a través del sistema de gestión.

Figura 8. Hoja de estilo XSL de visualización de datos.

Figura 9. Documento HTML generado por la transformación.

2) La interacción del médico con el paciente para la lectura de resultados de laboratorios o exámenes paraclínicos sería innecesaria en este sistema que propone un módulo estructural de la historia clínica electrónica (Interacción con Otros Niveles) en el cual los administradores de laboratorio (clínico y de imágenes diagnósticas) puedan ingresar sus resultados de los exámenes llevados a cabo al paciente. Con una sola consulta se pueda informar al médico la existencia de los mismos en la historia para que éste genere un tratamiento dentro del módulo de Tratamiento de la HCE, así el paciente puede estar informado de todo el proceso a través de Internet sin tener que desplazarse de su casa.

3) La arquitectura propuesta separa claramente el contenido de los documentos (clínicos, en este caso) de la presentación y de la lógica del negocio; este "layering" o división clara en niveles de compromiso por parte del sistema, favorece el posterior mantenimiento del mismo y permite enfocarse a la componentización de servicios en etapas posteriores.

4) La validación estructural de los contenidos de una historia clínica por medio de esquemas permitiría a entidades estatales (como el Ministerio de Salud) validar automáticamente la información que debe contener la historia clínica desde el punto de vista estructural. Aspectos semánticos que requieren ontologías específicas deben abordarse con mayor precaución y son tema de investigación actuales.

5) La integración de XML y sus tecnologías relacionadas (DOM,SAX, XSLT, entre otras), JSP, Java, HTTP, HTML, conectores a bases de datos y repositorios de datos brindan una plataforma de desarrollo de bajo costo para implementar aplicaciones de gran alcance como la aquí descrita.

6) La extensiones notacionales de UML para aplicaciones Web y la estrategia de diseño utilizada es bastante útil al modelar soluciones que involucren tecnologías Web.

7) Esta solución puede servir como patrón para soluciones de manejo de información de personas a través de la Web. Quedan abiertas muchas posibilidades de integración con otras soluciones en el mismo ámbito temático usando como "middleware" mecanismos de implementación de bastante auge actual, tales como XML-RPC´s.

REFERENCIAS

[1] Delgado, Miguel Antonio; Tolosa Peña, Diana Lucía."Aplicación para el almacenamiento y administración de historias clínicas sobre plataformas PC y agendas digitales personales con sistema operativo Windows CE" .Proyecto de Grado. Director: Diosa, Henry Alberto. Universidad Distrital Francisco José de Caldas. 2001

[2] www. java.programacion.net/tomcatintro/tomcat1.htm

[3] FENG,Ling et al." A semantic network-based design methodology for XML documents". ACM Information Systems. Vol. 20 Issue 4. October 2002. pages 390-421

[4] Isaza Peña, Carlos Andrés; Buenhombre González, Teresa."Gestión de Historias Clínicas Electrónicas Soportado en Tecnologías Web". Proyecto de Grado. Director: Diosa, Henry Alberto. Universidad Distrital Francisco José de Caldas. 2002

[5] DIOSA, Henry Alberto."Publicación Web de trabajos de investigación usando Lenguaje de Marcado Extensible ". Revista Ingeniería. Universidad Distrital Francisco José de Caldas. Vol. 6.No. 2. Año. 2001. págs. 35-41.

[6] CONALLEN, Jim. "UML Extensión for Web Applications".Addison-Wesley. Julio de 1999.

[7] Sun Microsystems: www.sun.com

Sitios en la Web:

- MYERS,T. Y NAKHIMOVSKY, A. "Professional Java XML Programming with Servlets and JSP". Ed. Wrox Press Ltda. 1999.
- World Wide WEB Consortium. Extensible Markup Language XML: www.w3.org/XML
- World Wide WEB Consortium. Extensible Stylesheet Language XSL: www.w3.org/Style/XSL
- World Wide WEB Consortium. XML-Schema: www.w3.org/TR/xmlschema-0
- World Wide WEB Consortium. Document Object Model(DOM) Level 3: www.w3.org/TR/DOM-Level-3-Core
- www.omg.org

Henry Alberto Diosa.
Ingeniero de Sistemas de la Universidad Nacional de Colombia, Msc. Teleinformática de la Universidad Distrital Francisco José de Caldas y docente de tiempo completo de la Facultad de Ingeniería de la misma Universidad. Coordinador Grupo de Trabajo CORBA-OSM. Realiza en la actualidad estudios de Doctorado en Ingeniería en la Universidad del Valle. E-mail: hdiosa@udistrital.edu.co

Carlos Andrés Isaza Peña.
Ingeniero de Sistemas de la Universidad Distrital Francisco José de Caldas. Se desempeña actualmente como Ingeniero de Desarrollo en la empresa Morris Technologies. E-mail: candresip@msn.com

Teresa Buenhombre González.
Ingeniera de Sistemas de la Universidad Distrital Francisco José de Caldas. Se desempeña actualmente como contratista del ITEC-Telecom dentro del Proyecto TARITEL. E-Mail: teresabg@msn.com