ESCUELA TÉCNICA SUPERIOR DE INGENIEROS

Titulación:

INGENIERO DE TELECOMUNICACIÓN

Título del proyecto:

Gestión de Informes de Alta con Tecnología Java y base de datos XML nativas.

Autor:

IRENE LAVADO GÓMEZ

Tutor:

ISABEL ROMAN MARTÍNEZ

DICIEMBRE DE 2003

A mi familia y a Fernando,

ÍNDICE

ÍNDICE 1

1- Motivación y objetivos 6

1.1- Introducción 6

1.2- Fases del proyecto 7

1.3- Estructura del documento 8

2- Introducción teórica 9

2.1- Situación de partida 9

2.1.1- Introducción 9

2.1.2- Informe de alta 10

2.2- JAVA 11

2.2.1- JAVA 11

2.2.2- Caracteristicas de los servlets 12

2.3- XML 13

2.3.1- Historia de XML 13

2.3.2- XML 15

2.3.3- XML Schema 17

2.3.4- DOM 22

2.3.5- XPath 27

2.3.6- Bases de datos nativas XML 29

2.4- Apis 36

2.4.1- dbXML 36

2.5- Conclusiones 37

3- Diseño e implementación 40

3.1- Software utilizado 40

3.1.1- Entorno Windows 40
3.1.2- JSDK 40
3.1.3- Sun ONE 41
3.1.4- Tomcat 44
3.1.5- Xindice Browser 44
3.2- Realización de la aplicación generadora de informes de alta 45
3.2.1- Introducción 45
3.2.2- Diagrama de flujo de la aplicación 46
3.2.3- Organización de los directorios 47
3.2.4- Estructura general del código 50
Código a destacar 51
Nuevo Paciente 54
Añadir/Modificar datos de un paciente(1) 57
Añadir/Modificar datos de un paciente(2) 58
4- Líneas de continuación 61
4.1- Líneas en el ámbito de la aplicación 61
4.1.1- Mejora del Informe de Alta 61
4.1.2- Seguridad 61
4.2- Líneas externas 62
4.2.1- Presentación de la información 62
4.2.2- Expansión de los esquemas XML 62
5- Conclusiones 64
6- Pliego de condiciones 65
6.1- Servlets 65
Action.java 65
AddAntecedentes.java 115

AddBioquimica.java 118
AddCircunstanciasDom.java 123
AddDemograficos.java 125
AddExploracion.java 130
AddHemograma.java 133
AddInforme.java 137
AddJuicio.java 139
AddJuicioComentarios.java 143
AddPatologia.java 146
AddProcedimiento.java 149
AddTratamiento.java 153
BuscaNodos.java 156
DBConnection.java 159
DeleteInforme.java 161
EditAntecedentes.java 163
EditBioquimica.java 166
EditCircunstancias.java 171
EditDemograficos.java 173
EditExploracion.java 177
EditHemograma.java 181
EditInforme.java 185
EditTratamiento.java 187
Group.java 190
Informe.java 195
ListInformes.java 197
ListJuicios.java 199
ListPatologias.java 204

ListProcedimientos.java 208
SearchInforme.java 212
ShowXML.java 216
Task.java 218
6.2- JSP 228
addAntecedentes.jsp 228
addBioquimica.jsp 241
addCircunstancias.jsp 260
addDemograficos.jsp 266
addExploracion.jsp 282
addHemograma.jsp 295
addInforme.jsp 310
addJuicio.jsp 314
addPatologias.jsp 322
addProcedimiento.jsp 327
addTratamiento.jsp 333
editPersonales.jsp 342
error.jsp 346
examenes.jsp 347
footer.jsp 352
header.jsp 353
header2.jsp 355
header3.jsp 359
index.jsp 361
listInformes.jsp 363
searchInforme.jsp 366
7 - Apéndice A 371

Bibliografía 371
Libros 371
Direcciones de internet 371
8 - Apéndice B 373
Instalación de xindice 373
9 - Apéndice C 375
W3c 375
¿Cómo trabaja el W3C? 375
Qué no es el W3C 376
Más información 376
10- Apéndice D 377

1- MOTIVACIÓN Y OBJETIVOS

1.1- INTRODUCCIÓN

Estamos en una época donde la información ha dejado de ser un privilegio, y el trabajo no se centra en la búsqueda, sino en la selección de información por lo que se hace necesario aplicar los conocimientos básicos de computación a diversos entornos, como la sanidad.

Como base para este proyecto se usó el presentado por Luis Corrales Vasco y titulado “Servidor Web para la generación de informes de alta basados en XML Schema”.

El objetivo de dicho proyecto fue diseñar una aplicación remota que permita al profesional médico (cliente) introducir los datos del Informe de Alta de un paciente de forma flexible. El servidor proporciona los formularios a rellenar, recoge la información introducida y, por último, la procesa con objeto de crear un documento XML válido con respecto a los esquemas XML construidos. Los esquemas XML se crearon siguiendo una serie de normas internacionales.

Este proyecto se diseñó con la tecnología PHP y XML, y los datos referentes a los pacientes se almacenaron en ficheros.

En este proyecto vamos a realizar una aplicación web del mismo informe médico, el de alta, pero enfocada desde otro punto de vista, ya que vamos a introducir el uso de base de datos nativas XML y vamos a usar la tecnología Java, en lugar de PHP. Como base de datos se ha elegido XIndice.

El objetivo principal en el diseño ha sido la claridad y simplicidad, de forma que la reutilización del código sea inmediata y no requiera de un estudio profundo. Para conseguir esto se han diseñado Servlets que desarrollan funciones simples, y se ha intentado que en las páginas JSP, haya el mínimo código posible.

1.2- FASES DEL PROYECTO

La ejecución del proyecto, a grandes rasgos, comenzó con un estudio del proyecto del que se partía, y del software a usar, después se pasó a profundizar en las tecnologías a usar y la base de datos elegida, y finalmente se diseñó la aplicación web.

Las fases temporales y de ejecución práctica han sido:

- 1 mes para el estudio del Proyecto “Servidor web para la generación de informes de alta basados en XML Schema”, y de los esquemas XML en él usado.
- 1 mes para el estudio y análisis de las posibilidades que ofrece el lenguaje de programación Java, y estudio de Servlets.
- 15 días para toma de contacto con el software a usar, en especial con Sun One, sobre el que se desarrolla todo el proyecto.
- 1 mes para el estudio de la base de datos elegida, XIndice, y del ejemplo base, Addressbook.
- 15 días para la construcción de una aplicación *web* autónoma, que usa la base de datos XIndice, y recoge y procesa una parte del informe de alta, los Datos Personales. La programación de la aplicación se desarrolló en Java y con la ayuda de HTML, y JSP.
- 15 días para el estudio del sistema DOM de estructuración de documentos XML, y aplicación de ésta a la web desarrollada.
- 1 mes y 15 días para extender la aplicación web a todos los datos del informe de alta.
- 1 mes de trabajo en la redacción de la presente Memoria del Proyecto.

1.3- ESTRUCTURA DEL DOCUMENTO

La estructura seguida en el desarrollo de esta memoria se basa en el siguiente esquema:

- Una introducción teórica donde se realiza un primer contacto con las necesidades de la informática en el ámbito sanitario, se discuten las tecnologías base susceptibles de ser aplicadas en nuestro ámbito y se expone el porqué de cada elección en particular.

- En segundo lugar se pasa al desarrollo de la aplicación web diseñada. Primero se comenta el software usado y algunas de las características de éste, y a continuación se pasa a comentar en detalle el código diseñado, las características de éste y el funcionamiento de la aplicación.

- A continuación se especifican las posibles líneas de continuación y conclusiones alcanzadas tras la finalización del diseño.

- Apartado dedicado al código fuente.

- Por último Apéndices, dedicados a la bibliografía usada y a detallar algunos de los proyectos comentados a lo largo de la memoria referentes a la colaboración existente entre la informática y la medicina.

2- INTRODUCCIÓN TEÓRICA

2.1- SITUACIÓN DE PARTIDA

2.1.1- Introducción

Estamos en una época donde la información ha dejado de ser un privilegio, y el trabajo no se centra en la búsqueda si no en la selección de información por lo que se hace necesario contar con los conocimientos básicos de computación aplicados a la medicina.

El SAS (Servicio Andaluz de Salud), como organismo sanitario público andaluz, está desarrollando políticas de salud que sitúan al ciudadano como eje fundamental del sistema sanitario de Andalucía.

Uno de los proyectos en los que trabaja el SAS es Diraya, cuya finalidad es obtener e implantar, mediante tecnologías informáticas, una Historia de Salud única para cada ciudadano en el ámbito de Andalucía. Inicialmente su ámbito es la Atención Primaria ya que, en gran medida, viene a sustituir las herramientas aportadas por el proyecto TASS, aprovechando la experiencia obtenida con el mismo. Se trata de dar un paso adelante dotando de integridad a los sistemas.

Diraya es un proyecto integrado por diversos servicios, entre los que podemos destacar:

- El empleo de la tarjeta sanitaria individual como sistema de identificación y acceso al sistema sanitario.
- La creación de la Base de Datos de Usuarios (BDU) para todos los ciudadanos andaluces.
- La puesta en marcha de una Historia digital de salud única o la oficina virtual Inters@s, que facilita que los ciudadanos realicen distintas gestiones sanitarias a través de internet.

La BDU se terminó en octubre de 2001 y contiene información básica de todos los andaluces, accesible en la red con el número de la tarjeta individual sanitaria.

Otro programa en el que el SAS está trabajando es Receta XXI, que también está integrado en el proyecto Diraya y afectará a los farmacéuticos de forma especial.

De toda la información sanitaria que puede existir de un paciente dentro de su historial médico nosotros centraremos nuestro proyecto en el Informe de alta.

2.1.2- Informe de alta

Es la parte del historial médico de un paciente que se realiza al finalizar el proceso asistencial de un paciente en el hospital.

La información contenida en el Informe de Alta debe ser, a grandes rasgos:

- *Datos personales del paciente*, a efectos de verificación de su identidad, lugar de residencia, formas de contacto y otros datos de interés.

- *Antecedentes médicos* de importancia tales como alergias, adicciones, patologías e ingresos previos, historia familiar, etc...

- *Datos de admisión* concretos como el nombre y tipo de hospital en el que ingresa el paciente, el agente sanitario remitente del informe o el encargado de los cuidados.

- *Circunstancias al ingreso* del paciente en el hospital: fecha, motivo, modo de ingreso...

- *Exámenes físicos* realizados al paciente durante su estancia en el hospital.

- *Juicio clínico* dictaminado por los profesionales sanitarios encargados de los cuidados.
- *Procedimientos realizados* a lo largo del período de convalecencia del paciente.
- *Tratamiento médico*: medicación prescrita por los médicos y forma adecuada de administración.
- *Circunstancias al alta*: fecha, tipo, y lugar al que se remite al paciente en el momento de recibir el alta médica.

2.2- JAVA

2.2.1- JAVA

Java es un lenguaje originalmente desarrollado por un grupo de ingenieros de Sun, utilizado por Netscape posteriormente como base para Javascript. Si bien su uso se destaca en el Web, sirve para crear todo tipo de aplicaciones (locales, intranet o internet).

Java es un lenguaje de objetos independiente de la plataforma.

Un objeto puede verse como una pieza de software que cumple con ciertas características como encapsulamiento y herencia.

Java implementa mecanismos de seguridad que limitan el acceso a recursos de las máquinas donde se ejecuta, especialmente en el caso de los Applets (que son aplicaciones que se cargan desde un servidor y se ejecutan en el cliente).

También está diseñado específicamente para trabajar en una red, de modo que incorpora objetos que permiten acceder a archivos en forma remota (via URL por ejemplo).

2.2.2- Caracteristicas de los servlets

La tecnología Servlet proporciona las mismas ventajas del lenguaje Java en cuanto a portabilidad (“write once, run anywhere”) y seguridad, ya que un Servlet es una clase de Java igual que cualquier otra, y por tanto tiene en ese sentido todas las características del lenguaje. Por otro lado los Servlets, una vez que son llamados por primera vez, quedan activos en la memoria del servidor hasta que el programa que controla el servidor los desactiva, de esta manera se minimiza en gran medida el tiempo de respuesta.

Además los Servlets se benefician de la gran capacidad de Java para ejecutar métodos en ordenadores remotos, conectar con bases de datos, etc. Se podría decir que las clases estándar de Java ofrecen resueltos muchos de los problemas que con otros lenguajes tiene que resolver el programador.

Además de las características citadas anteriormente, podemos destacar:

- Son independientes del servidor utilizado y de su sistema operativo.
- Los Servlets pueden llamar a otros Servlets, e incluso a métodos concretos de otros Servlets. De esta forma se puede distribuir de forma más eficiente el trabajo a realizar. Por ejemplo, se podría tener un Servlet encargado de la interacción con los clientes y que llamara a otro Servlet para que a su vez se encargara de la comunicación con una base de datos. De igual forma, los Servlets permiten direccionar peticiones de servicios a otros Servlets (en la misma máquina o en una máquina remota).
- Los Servlets pueden obtener fácilmente información acerca del cliente (la permitida por el protocolo HTTP), tal como su dirección IP, el puerto que se utiliza en la llamada, el método utilizado (GET, POST,…), etc.
- Permiten además la utilización de cookies y sesiones, de forma que se puede guardar información específica acerca de un usuario determinado, personalizando de esta forma la interacción cliente-servidor. Una clara aplicación es mantener la sesión con un cliente.

- Los Servlets pueden actuar como enlace entre el cliente y una o varias bases de datos en arquitecturas cliente-servidor de 3 capas (si la base de datos está en un servidor distinto).

- Asimismo, pueden realizar tareas de proxy para un applet. Debido a las restricciones de seguridad, un applet no puede acceder directamente por ejemplo a un servidor de datos localizado en cualquier máquina remota, pero el Servlet sí puede hacerlo de su parte.

- Al igual que los programas CGI, los Servlets permiten la generación dinámica de código HTML dentro de una página HTML. Así, pueden emplearse Servlets para la creación de contadores, banners, etc.

2.3- XML

2.3.1- Historia de XML

En los años sesenta, se perseguía la idea de estructurar los documentos de forma organizada, con el fin de facilitar el intercambio y la manipulación de los datos. IBM creó GML (Lenguaje de Marcado Generalizado) para satisfacer las necesidades de sus sistemas internos de edición. Empleó GML para producir libros, informes y otros documentos a partir de un solo conjunto de archivos fuente. En empresas específicas se introdujeron otras soluciones de estructuración de información, pero no se hacía nada por solucionar el tema a gran escala.

La primera tecnología de la información estandarizada y estructurada de cierta importancia fue SGML (Lenguaje de Marcado Generalizado Estándar), que también procedía de IBM. SGML se creó para proporcionar una manera de dar formato y mantener documentos legales en IBM. Posteriormente, se expandió y adaptó para ser utilizado en un amplio abanico de sectores como estándar de información de propósito general. No fue hasta 1986 que SGML emergió como estándar ISO. Aunque SGML es extremadamente potente, es muy complejo y se requiere una gran cantidad de software para procesarlo.

Debido a esta complejidad y a su elevado coste, SGML no supuso claramente una alternativa al hipertexto en los primeros días de Internet.

En 1989, Tim Berners-Lee y Anders Berglund, dos investigadores del Laboratorio Europeo de Física de Partículas (CERN), crearon un lenguaje basado en etiquetas para marcar documentos técnicos a fin de compartirlos en Internet. Este lenguaje fue finalmente ampliado en una aplicación simplificada de SGML llamada HTML, que supuso el primer formato de información estándar de la Web.

HTML tuvo mucho éxito y todo el mundo quería estar conectado. Cuando pasó la moda, tanto los particulares como las empresas empezaron a anhelar algo más que imágenes estáticas y texto. Lo que siguió fue una sucesión mareante de tecnologías tendentes a añadir interactividad a la Web. Algunas de estas tecnologías, como Java, han permanecido y florecido, mientras que otras se han quedado en la estacada. Aparte de las tecnologías interactivas, el propio HTML atravesó una rápida serie de mejoras (se añadieron más características, pero empezó a complicarse).

HTML fue originariamente diseñado como forma de presentar información estática sólo para fines de despliegue. Agregar interactividad a HTML, esperando que representara información dinámica, supuso el cambio de su propia naturaleza. Aún más importante que el cambio de información estática a dinámica fue el hecho de que se esperaba que HTML sirviera como forma de almacenar tipos específicos de datos, como transacciones financieras, catálogos de productos y resúmenes. Los desarrolladores web pronto se dieron cuenta de las ventajas de conectar páginas web a bases de datos de servidores, en cuyo caso se utilizaba HTML como interfaz de base de datos. HTML estaba cambiando de lenguaje de información basada en presentaciones a una tecnología de software.

HTML ha llevado a cabo una ímproba tarea satisfaciendo las necesidades de los desarrolladores web, considerando sus limitaciones innatas. Sin embargo, los miembros del Consorcio de World Wide Web (W3C) se dieron cuenta de que sería necesaria una alternativa mucho más amplia que HTML

para satisfacer las necesidades futuras de la Web. Quizás la limitación más patente de HTML sea su conjunto fijo de etiquetas. Es imposible que los desarrolladores web añadan etiquetas personalizadas con HTML, lo cual sería infinitamente más útil al tratar con datos pertenecientes a una aplicación o sector específicos. Por ejemplo, un editor de libros utilizaría mucho más etiquetas como <título>, <autor> e <isbn> que una etiqueta de párrafo <p> genérica.

En 1996, el W3C se propuso introducir el poder y la flexibilidad de SGML en el dominio de la Web. SGML ofrecía tres ventajas importantes que faltaban en HTML: extensibilidad, estructura y validación. Técnicamente, hay estructura en HTML, pero a la mayoría de los navegadores les trae sin cuidado si se adhiere a ella o no. SGML es mucho menos compasivo que los navegadores web a la hora de aventurarse fuera de la estructura permitida. Así, por razones puramente prácticas, HTML es un lenguaje poco estructurado.

El W3C presentó un equipo de expertos en SGML cuyo objetivo era el de crear una nueva tecnología de marcado con las ventajas nucleares de SGML y con la relativa simplicidad de HTML. El equipo “SGML para la Web” trabajó diligente y pacientemente en su nueva tecnología de marcado hasta febrero de 1998, fecha en la que el W3C lanzó la especificación XML 1.0.

2.3.2- XML

XML (eXtensible Markup Language) es un estándar industrial para representar datos, independiente del sistema. Al igual que **HTML** (HyperText Markup Language), XML encierra los datos en etiquetas, pero hay importantes diferencias entre los dos lenguajes de marcas. Primero, las etiquetas XML tienen relación con el significado del texto que encierran, mientras que las etiquetas HTML especifican cómo mostrar el texto encerrado. El siguiente ejemplo XML muestra una lista de precios con el nombre y el precio de dos cafés:

```
<priceList>
        <coffee>
```

```
            <name>Mocha Java</name>
            <price>11.95</price>
        </coffee>
        <coffee>
            <name>Sumatra</name>
            <price>12.50</price>
        </coffee>
</priceList>
```

Las etiquetas <coffee> y </coffee> le dicen al analizador que la información que hay entre ellas trata sobre un café. Las otras dos etiquetas dentro de las etiquetas <coffee> especifican que la información encerrada son el nombre del café y su precio por libra. Como las etiquetas XML especifican el contenido y la estructura de los datos que encierran es posible hacer cosas como archivar o buscar datos.

Una segunda diferencia importante entre XML y HTML es que las etiquetas XML son extensibles, permitiéndonos escribir nuestras propias etiquetas XML para describir nuestro contenido. Con HTML, estábamos limitados a usar sólo aquellas etiquetas que habían sido predefinidas en la especificación HTML.

Con la extensibilidad que proporciona XML, podemos crear las etiquetas que necesitemos para un tipo de documento en particular. Definimos las etiquetas usando un esquema de lenguaje XML. Un esquema describe la estructura de un conjunto de documentos XML y puede usarse para limitar los contenidos de los documentos XML.

Otras características también contribuyen a la popularidad de XML como un método para intercambio de datos. Está escrito en formato de texto, de forma que lo pueden leer tanto los humanos como los softwares de edición de texto. Las aplicaciones pueden analizar y procesar documentos XML, y los humanos también pueden leerlos en caso de que haya algún error en el procesamiento. Otra característica es que como un documento XML no incluye instrucciones de formateo, puede mostrarse de varias formas. Mantener los datos separados de

las instrucciones de formateo significa que los mismos datos pueden publicarse en diferentes medios.

Hay varias reglas básicas en XML. Echemos un vistazo a los objetivos establecidos por el W3C para XML:

- XML será directamente utilizable en Internet.
- XML soportará una amplia variedad de aplicaciones.
- XML será compatible con SGML.
- Será fácil escribir programas que procesan documentos XML.
- El número de características opcionales en XML debe mantenerse en un mínimo absoluto, idealmente cero.
- Los documentos XML deberán ser legibles por humanos y resultar razonablemente claros.
- El diseño de XML deberá ser preparado rápidamente.
- El diseño de XML será formal y conciso.
- Los documentos XML serán fáciles de crear.

XML se utiliza para crear otros lenguajes de marcas. Uno de los más conocidos es XSL. XSL está basado en XML. La relación entre XML y XSL es de suma importancia. Si XML son los datos en bruto, entonces XSL se usa para convertir esos datos en bruto en un formato de datos utilizable.

XML no sólo contiene datos, también puede contener la estructura e incluso la categorización de la información dentro del mismo documento. Con XML el documento lleva documentación acerca de sí mismo.

Microsoft Internet Explorer 4.0, 5.0, 5.5 y el 6.0 dan soporte a XML.

Netscape posee algún soporte para XML en Communicator/Navigator 6.0.

2.3.3- XML Schema

XML se ha consolidado definitivamente como un lenguaje estándar de información, tanto como medio de persistencia como de intercambio de información. En estas aplicaciones es de mucha relevancia qué posibilidades

nos ofrece esta tecnología de cara al modelado y la validación de la información.

En primer lugar es importante insistir en la necesidad, cada vez mayor, de poder estructurar correctamente, y de un modo más intuitivo, la información. Al igual que con la creación de la programación estructurada y posteriormente el almacenamiento relacionado de la información, actualmente está claro que ya no basta con almacenar la información identificando con que otros elementos está relacionada, sino que empieza a ser necesario, al igual que en el paradigma de la orientación a objetos, jerarquizar y estructurar la información.

Para proceder a esta estructuración dentro de un documento XML se han propuesto distintas soluciones dentro del entorno de W3C, todas enfocadas a definir un patrón externo al documento propiamente dicho que nos permita decir si el documento se adhiere a la estructura esperada o no.

Aquí vamos a tratar con dos de las principales soluciones, las **DTDs** y los **XML Schemas**, pero antes de iniciar la exposición de las diferencias entre ambas es importante echar un rápido vistazo a su origen.

Inicialmente el uso de las **DTDs** se basa en SGML, en el cual describían no sólo el vocabulario necesario para identificar todos los elementos de que iba a constar un documento, si no que también expresaba la estructura que dichos elementos debían respetar. Con la creación de XML, que recordemos que no es más que un subconjunto de SGML, fue necesario mantener la posibilidad de describir los elementos necesarios, al igual que en SGML, por ello se importaron las **DTDs** y todo su comportamiento.

Posteriormente, y debido al uso principalmente, se vio la necesidad de emplear otros métodos para describir esas necesidades inherentes a XML, con esta idea se creó XML Schema, se pretendía mejorar y ampliar la utilidad de las **DTDs.**

Como hemos indicado antes, una **DTD** es una especificación y estructuración necesaria que permite validar el contenido estructural y formal de un documento de SGML (y XML). Las **DTDs** se pueden usar para la

definición de modelos de contenido, es decir, en qué orden y qué elementos pertenecen a un elemento de orden superior en la jerarquía del documento; además permiten, aun que de modo muy limitado, imponer ciertas restricciones sobre el tipo de los elementos.

Ejemplo de DTD

```
<!ELEMENT Articulo (Cabecera, Contenido, Final)>
<!ELEMENT Cabecera (Titulo, Autor)>
<!ELEMENT Titulo ANY>
<!ELEMENT Autor ANY>
<!ELEMENT Contenido (Estracto, Cuerpo)>
.....
```

Ejemplo de XML Schema

```
<schema targetNamespace=”http://www.aqs.es/Schema_y_DTDs”
xmlns=”http://www.w3.org/2001/XMLSchema”
xmlns:AQS=”http://www.aqs.es/Schema_y_DTDs” >
<element name=”Articulo” type=”AQS:tArticulo” />
<complexType name=”tArticulo”>
<element name=”Cabecera” type=”AQS:tCabecera”/>
<element name=”Cuerpo” type=”AQS:tCuerpo”/>
<element name=”Final” type=”AQS:tFinal”/>f
        </complezType>
        <complexType name=”tCabecera”>
        <element name=”Titulo” type=”string”/>
        <element name=”Autor” type=”string”/>
        </complexType>
        .....
```

Al igual que las **DTDs**, los **Schemas** describen la estructura de la información. El motivo de la creación de este nuevo estándar para realizar la labor de las **DTDs** es, básicamente, la utilidad. Durante un tiempo, y a falta de

otra solución más ajustada, se emplearon los mecanismos que proporcionaba SGML para modelizar la información en XML. Pero el descubrimiento de nuevas aplicaciones de XML al margen de la estructuración de documentos forzó la creación de otras soluciones que ayudasen a solventar los nuevos problemas a los que se enfrentaba el mercado.

Dado que el surgir de una nueva tecnología donde ya existía otra es síntoma de que un grupo de gente cree que en el actual estado de ese campo o bien las cosas no están bien hechas o bien falta algo por hacer vamos a mostrar, que limitaciones tenían las DTDs para que fuese necesaria la aparición de XML Schema.

- Posee un lenguaje propio de escritura, lo cual deriva en problemas a la hora de:
    - el aprendizaje: no sólo hay que aprender XML si no que además hay que conseguir hacerse con el lenguaje de las **DTDs**,.
    - procesado del documento: las herramientas y parsers que se empleen para tratar los documentos de XML deben ser capaces de procesar las **DTDs** .
- No permite el uso de **namespaces**, estos permiten definir elementos con igual nombre dentro del mismo contexto, siempre y cuando se anteponga un prefijo al nombre del elemento, y lo más importante permiten el uso de elementos definidos en otros documentos.
- Tiene un tipado para los datos del documento extremadamente limitado, no permite definir el que un elemento pueda ser de un tipo numeral o de un tipo de fecha, sólo presenta variaciones limitadas sobre strings.
- El mecanismo de extensión es complejo y frágil, está basado en sustituciones sobre strings. Lo peor de dichas extensiones es que realmente no hace explicitas las relaciones en ningún momento, es decir, dos elementos que tienen definido el mismo modelo de contenido no presentan ninguna relación.

Estás pegas son superadas en la especificación de **XML Schema**, permitiendo un lenguaje mucho más expresivo. A parte de la expresividad, la

presencia de los Schemas permite un intercambio de información mucho más robusto.

A parte de solventar los problemas antes expuestos, **XML Schema**, permite una serie de ventajas adicionales que se consideraron importantes:

- Una estructura de tipos mucho más rica. En **XML Schema** se definen los tipos base que se pueden emplear dentro de esquema de XML, e.g., integer, bolean, string, date, etc.
- Permite tipos definidos por el usuario, llamados **Arquetipos**, dándoles un nombre y que se pueden emplear en distintas partes dentro del **Schema**.
- Es posible agrupar atributos, haciendo más comprensible el uso de un grupo de aspectos de varios elementos distintos, pero con denominador común, que deben ir juntos en cada uno de estos elementos.
- El trabajo con namespaces está especificado, permitiendo, dentro de la dificultad que conlleva trabajar con ellos, validar documentos con varios **namespaces**.
- Sin embargo, la característica que más resalta la utilidad de XML Schema es la posibilidad de extender **Arquetipos** de un modo específico, es decir permite lo que en términos de orientación a objetos se llama **herencia**. Considérese un esquema que extiende otro previamente hecho, se permite refinar la especificación de algún tipo de elemento para, por ejemplo, indicar que puede contener algún nuevo elemento del tipo que sea; pero dejando el resto del esquema antiguo completamente intacto.

Tal como hemos visto, es necesario empezar el *schema* definiendo los elementos más profundamente anidados dentro de la estructura jerárquica de elementos del documento XML. Es decir, tenemos que trabajar "de dentro hacia fuera".

2.3.4- DOM

El Modelo de Objetos de Documento, a diferencia de SAX, tiene su origen en el Consorcio World Wide Web (W3C). Mientras que SAX es un software de dominio público, desarrollado tras largas discusiones en la lista de distribución XML-dev, DOM es un estándar de la misma manera que lo es la especificación real de XML. DOM tampoco está diseñado específicamente para Java, sino para representar el contenido y el modelo de documentos en todos los lenguajes y herramientas de programación.

DOM se organiza en “niveles” en vez de en versiones. El Nivel Uno de DOM es una Recomendación aceptada y puede verse completa en http://www.w3.org/TR/RED-DOM-Level-1. El Nivel Uno detalla la funcionalidad y la navegación del contenido en un documento. Un documento en DOM no se limita a XML, puede ser HTML, o también otros modelos de contenido. El Nivel Dos añade al Nivel Uno módulos u opciones dirigidas a modelos específicos, como SML, HTML y Hojas de Estilo en Cascada (CSS).

La especificación DOM está diseñada para ser utilizada con cualquier lenguaje de programación. Por lo tanto, intenta utilizar un conjunto básico común de características disponibles en todos los lenguajes. La especificación DOM también intenta ser neutra en cuanto a las definiciones de interfaz. Esto permite a los programadores de Java aplicar sus conocimientos de DOM cuando trabajan con Visual Basic o Perl, y viceversa.

La especificación también trata todos los elementos del documento como un nodo, formado por un tipo y un valor. Esto proporciona un marco conceptual elegante para trabajar con todos los aspectos del documento.

El inconveniente de la neutralidad del lenguaje de DOM es que las metodologías y los patrones que se utilizan normalmente en los lenguajes de programación no se pueden emplear. Por ejemplo, en lugar de poder crear nuevos elementos (Element) mediante el conocido elemento new de Java, los programadores deben utilizar los métodos de construcción de fábrica. Los conjuntos de nodos se representan como listas de nodos (NodeList), en lugar de

los objetos normales List o Iterator. Estos pequeños inconvenientes se agregan a las prácticas de codificación poco habituales y al aumento de las líneas de código, y fuerzan al programador a aprender la metodología DOM en lugar de utilizar la forma más intuitiva.

DOM emplea una abstracción del tipo "todo integrado en un nodo". Esto significa que casi todos los elementos del documento XML, como por ejemplo Document, Element y Attr, heredan (extend) la interfaz Node. Este método no es sólo conceptualmente elegante, sino que permite que cada implementación distinta de DOM ponga al descubierto sus propias clases mediante las interfaces estándar, sin la pérdida de rendimiento que supone atravesar las clases de acomodación intermedias.

La abstracción "todo integrado en un nodo" pierde cierto valor debido al número de tipos de nodo existente y a la ausencia de uniformidad presente en los métodos de acceso. Por ejemplo, el método insertData se utiliza para establecer el valor de los nodos CharacterData, mientras que el valor de los nodos Attr (atributo) se establece mediante el método setValue. Mediante la presentación de diferentes interfaces para los distintos nodos, la uniformidad y la elegancia del modelo disminuyen y la curva de aprendizaje aumenta.

El DOM define varias interfaces, algunas de las cuales son más útiles que las otras. Cada interfaz posee un conjunto de métodos y atributos:

- **DOMImplementation**. Es una consulta al propio objeto y es independiente del documento que se le cargue. Devuelve información acerca del nivel del DOM que soporta el objeto.
- **DocumentFragment**. Es la parte ligera del documento.
- **Document**. Esta interfaz proporciona información acerca del propio documento principal que se ha cargado en el objeto. Posee varios métodos para crear nodos y atributos XML.
- **Node**. Todo en un documento puede ser considerado un nodo. Los elementos son nodos, los comentarios son nodos, y así sucesivamente. Esta interfaz contiene varios atributos generales apropiados para manipular cualquier tipo de nodo. De forma poco sorprendente, los

métodos y atributos del nodo se solapan con los de las demás interfaces. Por ejemplo, al tratar con un elemento, el atributo del nodo nodeName devolverá el mismo valor que el atributo del elemento tagName.

- **NodeList**. Es un conjunto ordenado de nodos, al que se puede acceder por medio de un índice.
- **NamedNodeMap.** Es una colección de nodos a los que se puede acceder por nombre. Los atributos de un elemento se almacenan como NamedNodeMap.
- **CharacterData.** Es una interfaz que trata sobre los datos de caracteres o con el texto de un documento.
- **Attr**. Esta interfaz trata sobre los atributos XML de un nodo. La versión abreviada Attr se usa para evitar la confusión con un atributo.
- **Element.** La mayoría de los nodos de un documento XML son elementos. Esta interfaz posee propiedades y métodos que sirven para tratar con los elementos y sus atributos XML.
- **Text.** Trata sobre el contenido de texto de un elemento. La mayor parte de la funcionalidad que requiere la maneja la interfaz CharacterData.

Vamos a examinar algunas de las interfaces examinando los atributos y métodos específicos, la interfaz Node, que es la interfaz clave del DOM. Todo en un documento puede considerarse un nodo, incluyendo el propio documento.

Atributos de sólo lectura de la interfaz de Node

Atributos	**Descripción**
nodeName	Proporcionan los nodos nodeName y nodeValue de todos los distintos tipos de nodos
nodeValue	
nodeType	Existen 12 tipos de nodos que se describen en el DOM del W3C. Entre ellos el que devuelve el nombre del tag, el que da el atributo, el que da el texto.
parentNode	Todos los nodos, a excepción de Documento, documentFragment, entidad, notación y Attr poseen un primario.
childNodes	Devuelve una NodeList de todos los nodos secundarios del nodo.
firstChild	Los nombres de estos atributos describen su valor de retomo. Éste es el objeto de nodoque contiene el relativo respectivo. Si la relación no existe, se devolverá null.
lastChild	
previousSibling	
nextSibling	
attributes	Devuelve un NamedNodeMap de todos los atributos
ownerDocument	Este atributo devuelve el objeto de documento como un nodo.

Métodos de la interfaz Node

Métodos	**Descripción**
insertBefore	Este método inserta el nuevo nodo secundario antes del nodo del secundario de referencia y devuelve new_node.
replaceChild	Este método sustituye al nodo secundario de referencia y devuelve el nodo sustituido.
removeChild	Este método elimina el nodo secundario de referencia y devuelve el nodo eliminado
appendChild	Este método adjunta el nuevo nodo al final de los demás nodos secundarios y devuelve el nuevo nodo
hasChildNodes	Este método es un valor booleano que comprueba si un nodo determinado tiene nodos secundarios.
cloneNode	El método cloneNodeO hará una copia del nodo donado. Si se clona un elemento, tomará un argumento true o false, que hará referencia a la profundidad del clónico.

La cuestión es cómo se incorpora un motor DOM a la aplicación:

- **Java**. Los programadores de Java son los que más fácil lo tienen. Registre el motor DOM para Java como clase y utilice todos sus métodos y propiedades.

- **C++.** El código fuente del analizador en C de IBM está disponible, y se puede llamar a las clases de este motor de forma directa. El motor de IBM también está disponible como DLL. El motor DOM de Microsoft sólo se puede usar como DLL o como objeto COM. Probablemente, la manera más fácil de incorporar código en C++ consista en usar Microsoft ADO como COM.
- **Visual Basic.** El motor DOM de Microsoft puede ejemplificarse en una aplicación de VB como objeto. Cuando se ha creado el objeto se puede acceder al documento por medio de todas las interfaces del DOM.

2.3.5- XPath

XPath (Lenguaje de camino XML) es una especificación por derecho propio, pero es usada principalmente por XSLT. La especificación XPath define cómo localizar un elemento específico dentro de un documento XML. Esto se logra haciendo referencia a nodos específicos en el documento XML; aquí, nodo se refiere a cualquier porción de datos XML, incluyendo elementos, atributos o datos de texto. En la especificación XPath, un documento XML se considera un árbol de estos nodos, donde se puede acceder a cada nodo especificando su posición en el árbol en que se encuentra.

XPath es un lenguaje no XML que se usa para dirigirse a partes de un documento XML. Es un lenguaje no XML, ya que no se describe para construcciones XML, como elementos y atributos, aunque se use para dirigirse a documentos XML. La sintaxis que emplea XPath está diseñada para ser utilizada en URI y valores de atributos XML, lo que significa que debe ser muy concisa. Ésta es la razón por la que XPath no se implementa como vocabulario XML. El nombre XPath se basa en el concepto de utilizar una notación de ruta para dirigirse a los documentos XML.

Para habilitar una forma de dirigirse a partes de un documento XML, XPath trata el documento como un árbol de nodos, lo cual es un modo bastante habitual de percibir los documentos XML. Los nodos de un documento XML pueden dividirse en nodos de elementos, nodos de atributos y nodos de texto. Algunos nodos tienen nombres, que pueden estar formador por un URI de

espacio de nombres opcional y un nombre local; un nombre que incluye un espacio de nombres se denomina expandido. A continuación se exponen los distintos tipos de nodos que pueden aparecer en un árbol XPath:

- Nodos raíz
- Nodos de elementos
- Nodos de texto
- Nodos de atributos
- Nodos de espacios de nombres
- Nodos de instrucciones de procesamiento
- Nodos de comentarios

Siempre hay un nodo raíz único que sirve como raíz de un árbol XPath y que aparece como su primer nodo. Todos los elementos de un documento tienen su correspondiente nodo de elemento que aparece en el árbol bajo el nodo raíz. Dentro de un nodo de elemento están todos los demás tipos de nodos, que se corresponden con el contenido del elemento. Los nodos de elementos pueden tener un identificador único asociado, que es útil a la hora de hacer referencia al nodo con XPath.

La premisa principal que subyace a XPath es el recorrido de un documento XML para llegar a un nodo determinado. Este recorrido se lleva a cabo por medio de expresiones. La especificación XPath describe cómo usar la sintaxis XPath para construir expresiones. Cuando se evalúan las expresiones XPath, acaban siendo un objeto de datos de uno de los tipos siguientes:

- node-set. Un conjunto de nodos
- boolean. Un valor true-false
- number. Un número de coma flotante
- string. Una cadena de texto

El modo en que se evalúa una expresión XPath depende completamente del contexto de la expresión, que no viene determinado por XPath. En otras palabras, ésta es la parte abstracta de XPath que se puede utilizar con Xpointer y XSLT para dirigirse a partes de documentos. El contexto de una expresión Xpath viene determinado por Xpointer o XSLT, que a su vez determina cómo se evalúa la expresión.

2.3.6- Bases de datos nativas XML

Durante los últimos años, XML se ha convertido en una de las más importantes tecnologías para la gestión de infraestructuras basadas en el Web.

La aceptación de esta tecnología se debe principalmente a dos de sus características: su capacidad como mecanismo de integración y la separación existente entre contenidos y presentación.

XML es un lenguaje idóneo para la construcción de sistemas integrados utilizando middleware orientado a mensajes o MOM. La alta extensibilidad y robustez de XML le convierte en la mejor solución para la integración de aplicaciones a través de mensajes, permitiendo una rápida adaptación a los continuos cambios en las necesidades de negocio, además de flexibilizar el intercambio de datos entre diferentes sistemas de información.

Por otro lado, la separación de los contenidos de su representación supone una mejora considerable en la gestión de la información. Los contenidos se crean y mantienen una sola vez reduciendo la multiplicidad de formatos de salida para los distintos dispositivos de visualización existentes en el mercado.

A pesar de las ventajas de las características comentadas, una última condición necesaria para el desarrollo definitivo de XML en la empresa es la del almacenamiento de la información. El dominio de XML sobre el intercambio de datos en el mercado depende de las capacidades de almacenamiento de las que se disponga. XML no puede ser una opción seria de negocio para el intercambio de información entre empresas si no se puede almacenar de forma eficiente. Los altos volúmenes de información gestionada hacen que las

conversiones de formato y el almacenamiento de datos en ficheros sean soluciones poco robustas y para nada eficientes.

La solución idónea para el almacenamiento de datos en formato XML es la de disponer de sistemas de gestión de información que sean capaces de gestionar estos datos en su formato nativo.

El almacenamiento de datos en formato XML nativo tiene diversas ventajas sobre la gestión realizada por las bases de datos relacionales convencionales. No es necesaria ninguna conversión de formatos y la estructura de los documentos se mantiene intacta. Los sistemas relacionales típicos no están dotados para la gestión jerárquica de datos que tiene XML, por lo que necesitan mecanismos de conversión a medida para el almacenamiento de información en este formato. Además tampoco disponen de mecanismos efectivos para contemplar el dinamismo de los documentos XML cuyo formato puede variar con facilidad.

Una de las soluciones más eficaces para almacenar datos en formato XML es la de las bases de datos nativas XML. Estas bases de datos proporcionan la funcionalidad necesaria para almacenar datos en formato XML y realizar consultas en la información gestionada. Están, por tanto, especializadas en almacenar datos en formato XML manteniendo el modelo XML asociado intacto.

Una base de datos nativa XML define un modelo lógico para cada documento XML y almacena y recupera información ajustada a ese modelo lógico. Como mínimo, estos modelos lógicos deben incluir elementos, atributos, secciones PCDATA y deben contemplar la ordenación de documentos.

Además, las bases de datos nativas XML tienen un documento XML como unidad fundamental de almacenamiento, al igual que una base de datos relacional tiene un registro que pertenece a una tabla determinada.

Otro aspecto a considerar es que no se requiere que tenga una estructura asociada para el almacenamiento físico de los datos. Puede ser construida sobre una base de datos relacional u orientada a objetos, o bien utilizar

formatos de almacenamiento propietarios como ficheros indexados o comprimidos.

En realidad, las bases de datos nativas XML no representan un nuevo modelo de base de datos de bajo nivel ni pretenden reemplazar a los sistemas de almacenamiento actuales. Se han planteado como una herramienta de soporte al desarrollo que permitan a los desarrolladores disponer de una solución robusta para el almacenamiento y manipulación de datos en formato XML.

2.3.7.1 – Almacenamiento de XML nativo

Las bases de datos nativas XML almacenan los documentos creando modelos lógicos. Estos modelos se basan directamente en XML o en alguna de las tecnologías relacionadas como DOM o Infoset. Estos modelos incluyen diferentes niveles de agregación y complejidad, así como un soporte completo para la gestión de datos semi-estructurados.

Los modelos son automáticamente mapeados por las bases de datos XML nativas al mecanismo de almacenamiento correspondiente. El mapeo utilizado por estas bases de datos garantizará la correcta utilización del modelo asociado al documento original.

Una vez que se ha almacenado la información, y con el objeto de aprovechar al máximo la capacidad de estos sistemas, es necesario seguir utilizando herramientas XML que permitan realizar otros tipos de operaciones.

La principal ventaja de las bases de datos nativas XML es la abstracción. El programador no tiene porqué conocer los detalles de cómo se almacena la información, pudiendo construir sus aplicaciones con total libertad.

2.3.7.2 – Dependencia del esquema

Las bases de datos nativas XML gestionan los documentos como si fueran colecciones de datos, permitiendo realizar consultas y manipulaciones sobre esos documentos en forma de conjuntos de información. Es un concepto similar al de las tablas en las bases de datos relacionales.

La diferencia con los sistemas relacionales es que no todas las bases de datos nativas XML necesitan un esquema para generar una colección de documentos. Esto quiere decir que se puede almacenar cualquier documento XML en la colección independientemente de su estructura, permitiendo la realización de consultas en todo el conjunto de datos con una gran flexibilidad.

Las bases de datos nativas XML que soportan esta característica se denominan independientes del esquema.

Disponer de colecciones de documentos independientes del esquema proporciona a la base de datos una alta flexibilidad y facilita el desarrollo de aplicaciones. Por otro lado, esta característica dificulta la tarea de los administradores de base de datos debido a posibles problemas de integridad de datos.

En ocasiones es preferible disponer de menos flexibilidad y asegurar una correcta integridad referencial, debiendo decantarse por soluciones dependientes del esquema u otro tipo de aplicaciones de almacenamiento de documentos XML.

2.3.7.3 – Capacidades de consulta

En un principio, XPath es el lenguaje de consulta estándar para bases de datos nativas XML. Así, y con el objeto de facilitar la realización de consultas en bases de datos, se ha modificado el estándar inicial para dotar a XPath de capacidades de consulta en colecciones de documentos.

En la actualidad se está desarrollando un lenguaje de consulta más orientado a bases de datos XML llamado XQuery. Aunque no hay una versión estándar del lenguaje, algunos fabricantes ya proveen prototipos del mismo en sus implementaciones.

Para optimizar la realización de consultas, las bases de datos nativas XML ya disponen de mecanismos de indexado que aumentan las capacidades de búsqueda en colecciones de documentos de gran tamaño. Los detalles de implementación de estos mecanismos de indexado pueden variar entre los

distintos fabricantes, pero casi todos ellos proveen de mecanismos de este tipo para la optimización de las consultas.

2.3.7.4 - XIndice

XIndice es una base de datos desarrollada en Java que se ha diseñado para el almacenamiento de grandes cantidades de pequeños documentos XML. Se distribuye en forma de código abierto por la Apache Software Foundation.

Puede indexar tanto valores estructurales como contenidos, almacenando los documentos XML en formato comprimido para ahorrar espacio en disco.

Los documentos se estructuran en una jerarquía de colecciones y pueden ser consultados utilizando XPath. Como lenguaje de actualización se puede emplear XUpdate, y para la gestión de enlaces entre documentos se utiliza un lenguaje experimental similar a XLink.

Soporta tres tipos de interface de programación: XML:DB, un API para CORBA y un plugin XML-RPC que soporta accesos desde PHP, Perl y Applescript. La funcionalidad del servidor es extensible utilizando la tecnología XMLObjects.

2.3.7.5 – Estudio comparativo

Nombre	Licencia	Sistema Almacen.	T	E	C	I[1]	A	J	Acceso	API	Otros
4Suite	Open Source	Orientado a Objetos	Sí	¿?	Sí	Sí	Sí	Sí	HTTP, RPC, FTP, CORBA, SOAP	Python	SAX, DOM, XSLT, XInclude, XPointer, XLink, XPath, RDF, XUpdate
Birdstep RDM XML	Comercial	Orientado a Objetos	Sí	No	No	Sí	No	Sí		C, C++, Java	SAX, DOM, XPAth, Dispositivos móviles
Centor Interaction Server	Comercial	Solución Propietaria	¿?	Sí	Sí	Sí	Sí	¿?	JDBC, ODBC, EJB	C++, Java	Motor de workflow y estilos. Gestor de presentación
Cerisent XQE	Comercial	Solución Propietaria	Sí	¿?	Sí	¿?	¿?	¿?			XQuery, XPath, XML schemas, SGML, HTML
Coherty XML DB	Comercial	Solución Propietaria	Sí	Sí	Sí	Sí	Sí	Sí	HTTP	Java, JSP	XPath, XSL, XSQL, XML/SQL, DOM
DBDOM	Op. Source	Relacional	Sí	Sí	Sí	Sí	Sí	Sí	JDBC	Java	DOM, PosgreSQL
dbXML	Comercial	Solución Propietaria	¿?	Sí	Sí	Sí	Sí	¿?	HTTP, XML/RPC	Java, Javascript, JSP, Python	XPath, XSLT, ViXen, XUpdate, XQuery

[1] T: Transaccionalidad; E: Eventos o Triggers; C: Concurrencia; I: Indexado; A: Control de Acceso; J: Enlazado de datos

Nombre	Licencia	Sistema Almacen.	T	E	C	I[2]	A	J	Acceso	API	Otros
DOM-Safe	Comercial	Sistema Propietario	Sí	¿?	Sí	¿?	¿?	¿?	HTTP, WebDAV		XPath, XLink, SAX2, DOM
eXist	Open Source	Relacional	¿?	¿?	Sí	Sí	Sí	¿?	HTTP, XML/RPC, WebDAV	Java	XPath, Oracle, MySQL, PostgreSQL
eXtc	Comercial	Sistema Propietario	Sí	¿?	Sí	¿?	Sí	Sí	HTTP, SOAP, WSDL		DOM, SQL, XPath
eXtensible Information Server	Comercial	OO, relacional y otros	Sí	Sí	¿?	Sí	¿?	Sí	COM, .NET	Java, VisualBasic	XConnects, XPath, XQuery, Updategrams
GoXML DB	Comercial	Sistema Propietario	Sí	¿?	Sí	Sí	Sí	Sí	JDBC, GoXML DB Explorer, WebDAV	Java, scripts	XQuery, DRDs, XML schemas, MD5
Infonyte DB	Comercial	Sistema Propietario	Sí	¿?	Sí	Sí	Sí	Sí	HTTP	Collection, JSP, Java, Servlets	DOM, XQL, XPath, XSLT
Ipedo XML Database	Comercial	Sistema Propietario	Sí	¿?	Sí	Sí	Sí	Sí	SOAP, .NET, COM, EJB, WSDL, HTTP	Java	XQuery, DTDs, XML schemas, XPath, XSLT, DOM

[2] T: Transaccionalidad; E: Eventos o Triggers; C: Concurrencia; I: Indexado; A: Control de Acceso; J: Enlazado de datos

Nombre	Licencia	Sistema Almacen.	T	E	C	I[3]	A	J	Acceso	API	Otros
Lore	Invest.	Semi Estr.	¿?	¿?	Sí	Sí	Sí	¿?			DataGuide
Lucid XML DM	Comercial	Propietario	Sí	¿?	Sí	Sí	Sí	¿?		Java	XPath
MindSuite XDB	Comercial	OO	Sí	Sí	Sí	Sí	Sí	¿?	CORBA	Java, C++	DOM, XPath, XSLT
Natix	Comercial	Ficheros	No	¿?	Sí	Sí	Sí	¿?		Java, C++	XPath
Neocore XML Management System	Comercial	Sistema Propietario	Sí	¿?	Sí	Sí	Sí	¿?	HTTP, EJB, HTTPS, COM, .NET	Java, C++	XPath, X-Aware
Ozone	Open Src.	OO	Sí	No	Sí	¿?	Sí	¿?	RMI	Java	DOM
Sekaiju	Comercial	Sistema Propietario	Sí	¿?	Sí	Sí	Sí	¿?	HTTP, COM, SOAP	Visual Basic	RelaxNG, Xbatch
SQL/XML-IMDB	Comercial	Sistema Propietario	Sí	¿?	Sí	Sí	Sí	¿?	.NET, HTTP	C++, PHP, VBasic, Delphi, Perl	XQuery, SQL92
Tamino XML Server	Comercial	Sistema Propietario	Sí	¿?	Sí	Sí	Sí	¿?	JDBC, ODBC, SOAP, WebDAV	JSP	X-Tension, XPort, DataMap, DOM, JDOM, SAX, XSLT, XQuery, CSS, XPlorer, XNode, XApplication

[3] T: Transaccionalidad; E: Eventos o Triggers; C: Concurrencia; I: Indexado; A: Control de Acceso; J: Enlazado de datos

Nombre	Licencia	Sistema Almacen.	T	E	C	I[4]	A	J	Acceso	API	Otros
TeamXML	Comercial	Sistema Propietario	¿?	¿?	Sí	Sí	Sí	Sí	EJB, SOAP, .NET	OpenAPI (Java)	TeamSite
TeraText DBS	Comercial	Sistema Propietario	¿?	¿?	Sí	Sí	Sí	¿?	SOAP, WebDAV, LDAP	Java, C++	SGML, Unicode, PDF, Z39.50, MsWord
TEXTML	Comercial	Sistema Propietario	Sí	¿?	Sí	Sí	Sí	¿?	COM, COM+, ASP, WebDAV, .NET	Java, C++	MTS, SAP, Lotus, DataExchanger
TigerLogic XDMS	Comercial	Sistema Propietario	Sí	¿?	Sí	Sí	¿?	¿?	SOAP, HTTP, JCA	Java	XPath, XSLT, DTD, XML schemas, XA
TOTAL XML	Comercial	Objeto-Relacional	¿?	¿?	Sí	¿?	Sí	¿?	JDBC, ODBC	Java	DTD, SQL, XPath, SAX, StrivaDETAIL, DOM
Virtuoso	Comercial	Propietario (Relacional)	Sí	Sí	Sí	Sí	Sí	Sí	JDBC, ODBC, HTTP, SMTP, LDAP, POP3, OLE-DB, J2EE, WebDAV, SOAP, .NET		XA, XPath, XQuery, XSLT, SQL92

[4] T: Transaccionalidad; E: Eventos o Triggers; C: Concurrencia; I: Indexado; A: Control de Acceso; J: Enlazado de datos

Nombre	Licencia	Sistema Almacen.	T	E	C	I[5]	A	J	Acceso	API	Otros
XDB	Open Source	Relacional	¿?	¿?	¿?	¿?	¿?	¿?		C++	XPath, DOM, PosgreSQL, SAX, DTD, XML schemas, XMLQuery,
XDBM	Open Source	Sistema Propietario	¿?	¿?	¿?	¿?	¿?	¿?			DOM
X-Hive/DB	Comercial	OO y Relacional	Sí	¿?	Sí	Sí	Sí	Sí	JDBC, WebDAV	Java	XQuery, XPath, DOM, XSLT, Xlink, XPointer
XIndice	Open Source	Sistema Propietario	No	No	¿?	Sí	¿?	Sí	XML:DB, CORBA, XML-RPC	Java	XPath, Xlink, XUpdate, XMLObjects
Xyleme Zone Server	Comercial	Sistema Propietario	¿?	Sí	Sí	Sí	Sí	¿?	SOAP	Java, C++	XQuery,

[5] T: Transaccionalidad; E: Eventos o Triggers; C: Concurrencia; I: Indexado; A: Control de Acceso; J: Enlazado de datos

2.4- APIS

2.4.1- dbXML

La idea principal del dbXML es proporcionar una manera simple de salvar y de manejar una gran cantidad de documentos XML. Esto se logra salvando documentos en colecciones, donde cada documento individual se salva en una forma pre-analizada y comprimida. Esto incrementa notoriamente al velocidad al trabajar con datos XML.

Almacenar documentos en colecciones proporciona un mecanismo fácil para preguntar y manipular los documentos como conjunto. Esta Api presenta una gran flexibilidad para el manejo de documentos mixtos, incluso mezclar los documentos de esquemas totalmente diversos en la misma colección.

DBXML es un sistema de gerencia de datos diseñado específicamente para las colecciones de documentos XML. Se incluye fácilmente en aplicaciones existentes, altamente configurable, y extensible.

La compresión de XML permite que los documentos sean salvados en forma de tokens, permitiendo un funcionamiento rápido y la conservación de la memoria.

Los documentos se ocultan en memoria para mejorar su funcionamiento. El almacenamiento en memoria inmediata se puede controlar mediante la colección y los niveles del documento.

El dbXML incluye:

- Almacenamiento de las colecciones de documentos XML.
- Motor de Base de datos multihilo, que optimiza los datos XML.
- Almacenamiento comprimido pre-analizado del documento.
- Motor XPath query.
- Índices de la colección para mejorar el rendimiento de las consultas.

El servidor del dbXML viene con un conjunto de las herramientas de la línea de comandos que permiten realizar todas las funciones básicas de administración.

DBXML proporciona además el XUpdate, el cual proporciona un lenguaje simple para actualización de documentos XML. Permite declarar específicamente qué cambios deben ser realizados.

2.5- CONCLUSIONES

Las aplicaciones Web de la actualidad se enfrentan con un número abundante de problemas que ni siquiera se habían tenido en cuenta hace diez años. Los sistemas, distribuidos a lo largo de kilómetros, deben funcionar rápidamente y de manera intachable. Los datos, provenientes de sistemas heterogéneos, bases de datos, servicios de directorio y aplicaciones, deben transferirse sin perder ni un solo decimal. Las aplicaciones deben ser capaces de comunicarse no sólo con otros componentes de negocio, sino con todo un

conjunto de sistemas de negocio, a menudo entre diferentes compañías y tipos de tecnología. Los clientes ya no están limitados a clientes de tipo “pesado”, sino que pueden ser navegadores de web con soporte de HTML, teléfonos móviles con soporte de WAP (Protocolo de Aplicación Inalámbrico) u ordenadores de bolsillo con lenguajes de marcado totalmente diferentes. Los datos y la transformación de éstos, se han vuelto la pieza central decisiva de todas las aplicaciones desarrolladas en la actualidad.

XML ofrece el medio para que los programadores cumplan todos estos requisitos. **XML** hace los datos portables. Por otro lado la plataforma **Java** hace el código portable. Y los APIs Java para XML hacen fácil el uso de XML. Si ponemos todo esto junto tenemos la combinación perfecta: **portabilidad de datos, portabilidad de código y facilidad de uso**. De hecho, con los APIs de Java para XML, podremos obtener los beneficios del XML con un uso directo de XML muy pequeño.

XML se reconoce hoy en día como uno de los lenguajes de mayor impacto para la integración de servicios a través de la web. Es por eso que las empresas están descubriendo rápidamente los beneficios de usar XML para la integración de los datos, tanto internamente para compartir datos legales entre departamentos, como externamente para compartir datos con otras empresas. Y a causa de la integración de datos que ofrece XML, se ha convertido en indispensable para los nuevos proyectos relacionados con la Web.

Las arquitecturas de aplicaciones distribuídas hacen uso extensible de este lenguaje como mecanismo integrador entre los diferentes niveles de una aplicación. Desde esta perspectiva, la inmensa mayoría de documentos XML son creados por programas y procesados por otros para posteriormente ser distribuídos.

La parte realmente dura del desarrollo de servicios Web es la programación de la infraestructura, o "tuberías", como las capacidades de seguridad y de mensajería, el manejo de transacciones distribuidas, y el control de almacenes de conexiones. Otra dificultad es que esos servicios web deben ser capaces de manejar un enorme número de usuarios simultáneamente, por

eso las aplicaciones deben ser altamente escalables. Estos requerimientos son los que ofrece exactamente la plataforma Java TM, Enterprise Editon (J2EE TM). Si añadimos a esto que la plataforma J2EE es una tecnología demostrada con múltiples vendedores ofreciendo productos compatibles a día de hoy, nos encontramos con que la plataforma J2EE es la mejor plataforma para desarrollar servicios Web. Y con los nuevos APIs de Java para XML, el desarrollo de servicios Web se está simplificando cada vez más.

3- DISEÑO E IMPLEMENTACIÓN

3.1- SOFTWARE UTILIZADO

3.1.1- Entorno Windows

El software instalado para la realización del proyecto es:

- JAVA2SDK v1.4.1_02
- Sun ONE Studio 4, update 1, Edition. Se usó tambien la Enterprise Edition pero al ser una versión de prueba no se usó en todo el proyecto.
- Apache TOMCAT 4.1.22 y 4.1.24 Aunque el Sun ONE posee una versión del Tomcat se instaló una versión más moderna para evitar posibles problemas en las ejecuciones. Se probaron los 2 versiones del Tomcat.
- Xml-xindice-1.0. Base de datos usada.
- XIndice-Browser. Explorador que permite de forma sencilla y clara crear, borrar y modificar las colecciones de documentos almacenados en XIndice.

3.1.2- JSDK

El JSDK (Java Servlet Developer Kit), distribuido gratuitamente por Sun, proporciona el conjunto de herramientas necesarias para el desarrollo de Servlets.

El JSDK consta básicamente de 3 partes:

- El API (Application Programming Interface) del JSDK, que se encuentra diseñada como una extensión del JDK propiamente dicho. Consta de dos paquetes que se encuentran contenidos en

javax.servlet y javax.servlet.http. Este último es una particularización del primero para el caso del protocolo HTTP. Mediante este diseño lo que se consigue es que se mantenga una puerta abierta a la utilización de otros protocolos que existen en la actualidad (FTP, POP, SMTP, etc.), o vayan siendo utilizados en el futuro. Estos ficheros están almacenados en un fichero JAR (\lib\jsdk.jar).

- La documentación propiamente dicha del API y el código fuente de las clases.

La aplicación servletrunner, que es una simple utilidad que permite probar los servlets creados sin necesidad de hacer complejas instalaciones de servidores HTTP. Es similar en concepción al appletviewer del JDK.

3.1.3- Sun ONE

La Enterprise Edition for Java ofrece:

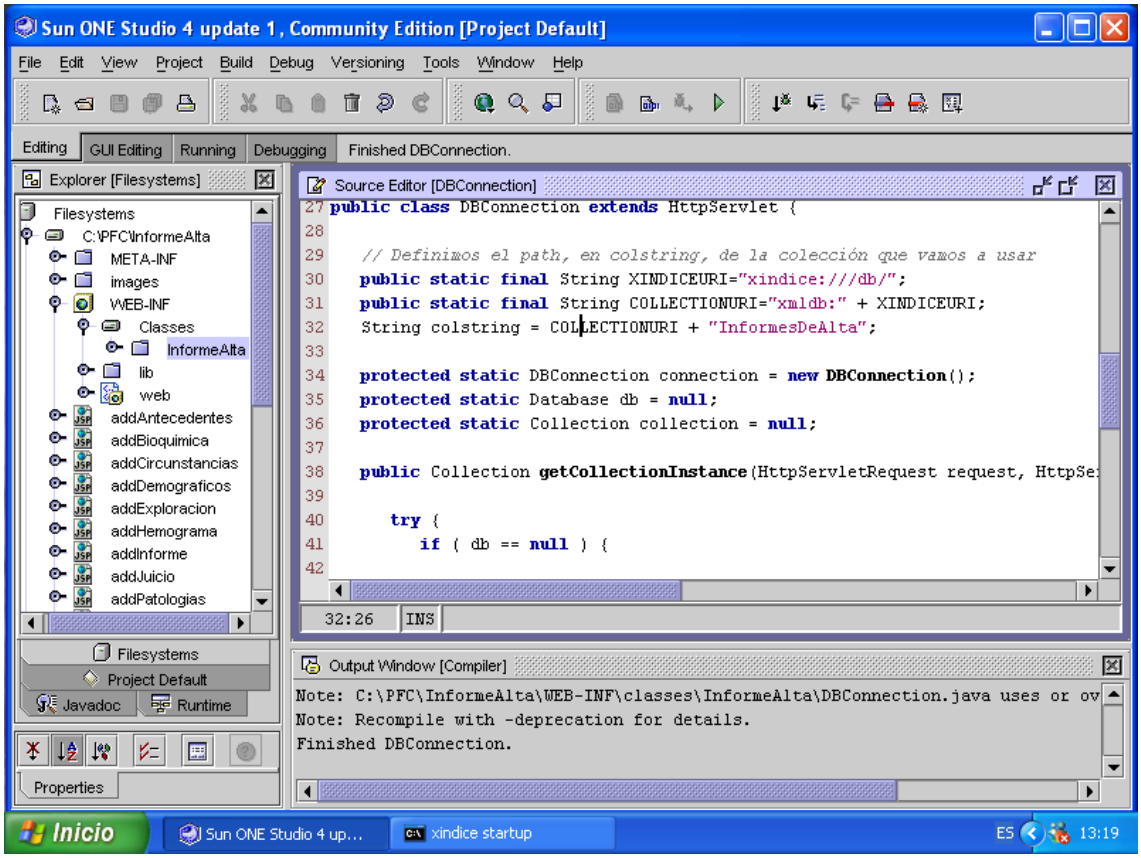
- Desarrollo, incorporación e implementación aplicaciones de componente Java 2 Platform, Enterprise Edition (J2EE[tm]), así como aplicaciones de servicios web basadas en estándares, en Sun ONE Application Server y otras plataformas líderes en el mercado.

- Admite el desarrollo a la especificación J2EE 1.3. Para desarrollar las aplicaciones de componente J2EE 1.2, usar el software Sun ONE Studio 3.0.

La Community Edition, por su parte, ofrece:

- Crear componentes JavaBeans[tm], subprogramas y aplicaciones independientes, así como aplicaciones web de reconocimiento de base de datos.

- Desarrollar componentes JavaBeans, JavaServer Pages[tm], (JSP[tm]), Servlet y JDBC[tm].



Algunas de las características más importantes del Sun ONE Studio4 update1 son:

- Gratuito, sólo el Sun ONE Studio 4 Community Edition.
- Posee las últimas tecnologías Java: La plataforma Java es la fundación para la creación y el desarrollo de servicios de red. Su flexibilidad hace de él la plataforma ideal para el desarrollo de servicios en demanda hoy y mañana. El IDE de Sun ONE Studio 4 update 1 es compatible con J2EE 1.3, y permite construir aplicaciones de alta seguridad, abiertas, robustas, viables y flexibles con soporte para los últimos estándares para Java 2 platform, Java 2 Swing/Java Foundation Classes (JFC), Java2D, Java Collections, Java JAX Pack, APIs de acceso y mucho más. Debido a que la tecnología Java es independiente de la plataforma, no habrá que migrar la aplicación a otra plataforma, lo que reduce el coste. Permite disfrutar de los beneficios de Java "Write once, Run anywhere."

- Rapidez en el desarrollo de código de calidad: Sun ONE Studio IDE incluye wizards, templates, y otros productos que autogeneran código y realizan tareas, además permite completar y desarrollar las aplicaciones en menos tiempo. Posee muchas herramientas, como, editor de lenguaje Java, editor GUI, debugger, compilador, ayuda online, interoperabilidad para objetos orientados al desarrollo con JNDI, CORBA, RMI, LDAP, internacionalización, explorador Web integrado, y gestión de proyectos, son algunas de las posibilidades de creación y montaje de componentes Java. Construir el código es solo la primera parte del trabajo. Debugging el código asegura una alta calidad y robustez en las aplicaciones. El avanzado y sofisticado debugger visual del que dispone permite acortar la fase de pruebas y debugging del desarrollo.
- Racionalizar el desarrollo de contenidos web dinámicos: Racionaliza el proceso de la creación de contenidos web dinámicos con soporte JSP para la rápida creación de aplicaciones web, Servlets y páginas JSP. También dispone de librerías de etiquetas para las presentaciones y acceso a las bases de datos, incluye soporte para Java Standards Tag Library (JSTL). Tiene un poderoso editor de JSP que permite crear tu propia librería de etiquetas. Sofisticado debugging que permite ejecutar paso a paso el código de los JSP o Servlet, y el monitor de ejecución permite ver las transacciones HTTP entre cliente y servidor.
- Cantidad de soporte técnico gratuito: Sun dispone de un portal dedicado al desarrollo, Sun ONE Studio Developer Resources, donde se puede encontrar de forma gratuita, soporte, artículos técnicos, "how-to", forums de desarrolladores, código de ejemplos, descargas de último software, y mucho más.

Características clave y ventajas	
Características	**Ventajas**
Desarrollo y publicación de servicios web a partir de componentes EJB existentes y ponerlos a disposición de los clientes que se comunican utilizando XML a través de HTTP. Integración del IDE en un servidor de aplicaciones para distribuir servicios web a otros clientes web, entre otras muchas funciones. (sólo Enterprise Edition for Java)	Cree aplicaciones de servicios web con herramientas de reconocimiento de servicios.
Crear contenido Web dinámico utilizando la tecnología JSP/Servlets	Desarrolle, pruebe e implemente aplicaciones web con la tecnología JavaServer Pages utilizando bibliotecas de etiquetas preempaquetadas, entre otras muchas funciones.
Integración perfecta con Sun ONE Studio 7, Compiler Collection. (sólo Enterprise Edition for Java)	Mayor productividad del desarrollador gracias a una interfaz de usuario única para varios idiomas.
Plantillas para la creación de MIDlet y paquetes de MIDlet. (sólo Mobile Edition)	Reduce el tiempo necesario para crear aplicaciones MIDlet.
Preparado para integrarse con emuladores/kits de desarrollo de software de terceros a través de la interfaz Unified Emulator Interface. (sólo Mobile Edition)	Mejora considerablemente la capacidad de los fabricantes de dispositivos y proveedores de emuladores para integrar kits de desarrollo de software con el IDE de Sun ONE Studio.

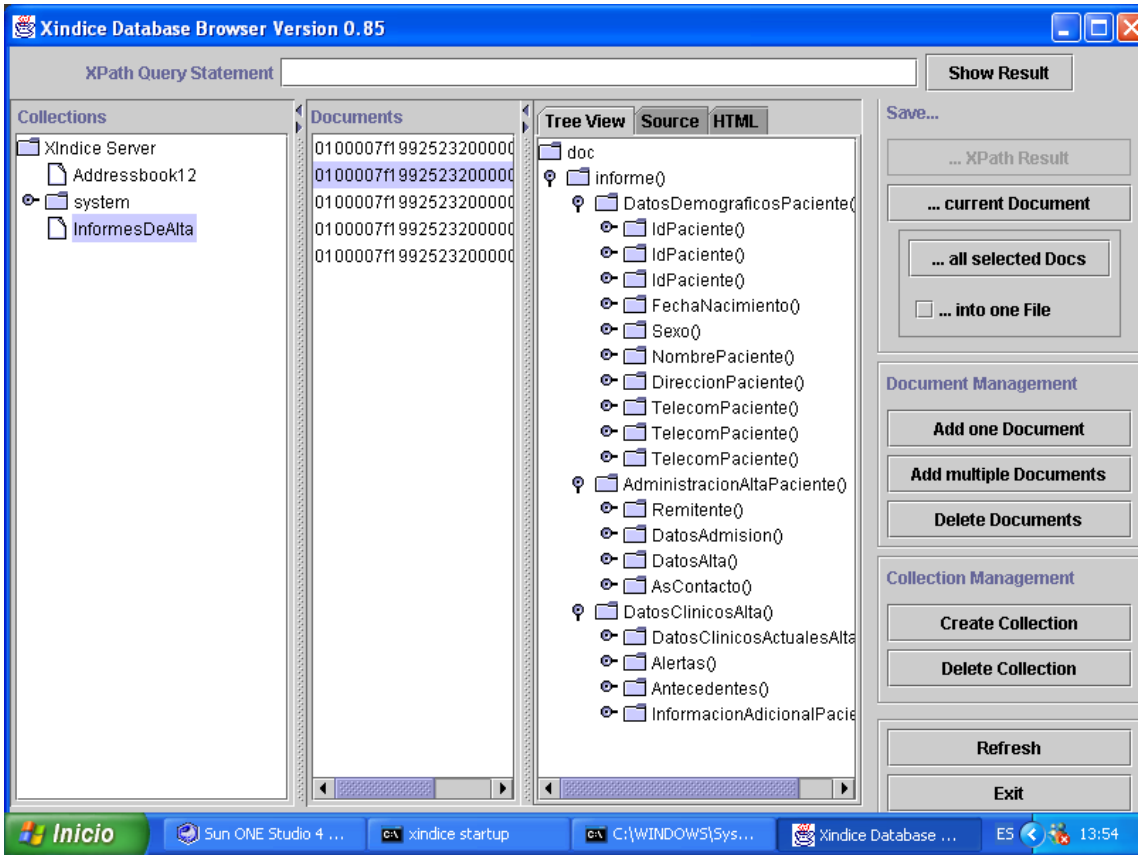
3.1.4- Tomcat

El servidor Tomcat es una aplicación web basada en Java creada para ejecutar servlets y páginas JSP.

3.1.5- Xindice Browser

Este browser permite ver las colecciones creadas en la base de datos, y los documentos que las componen. Es de gran utilidad, ya que además de esto permite, añadir o borrar, tanto colecciones como documentos.

Desde él también se pueden realizar búsquedas Xpath.



3.2- REALIZACIÓN DE LA APLICACIÓN GENERADORA DE INFORMES DE ALTA

3.2.1- Introducción

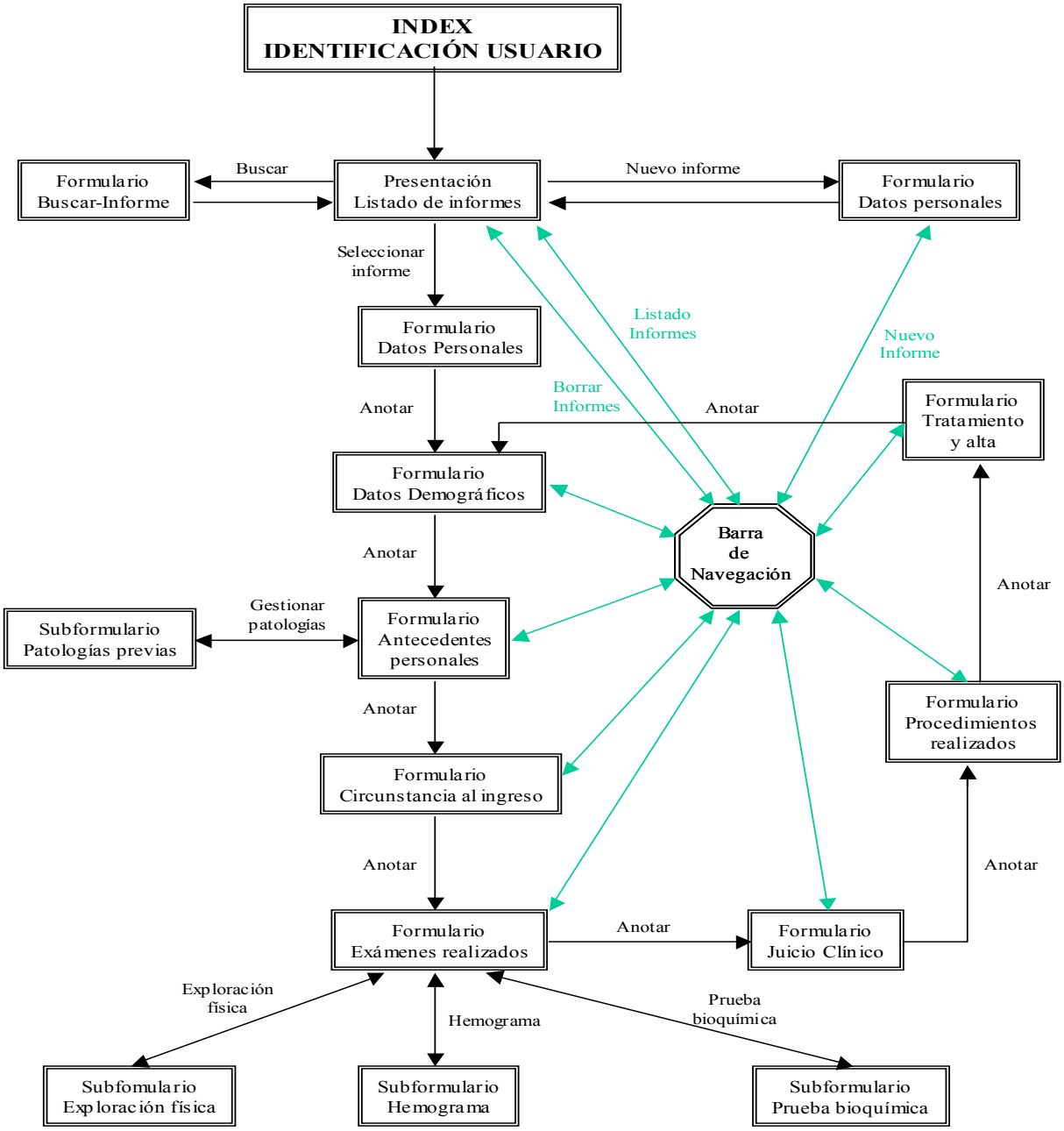
Buscamos el diseño de una aplicación remota, que permita al profesional de la medicina (cliente) introducir, modificar o borrar los datos del paciente, que en este caso harán referencia al informe de alta. El servidor, por su parte, debe proporcionar los formularios necesarios para dichas tareas, deberá recoger la información y procesarla, para crear el documento XML que se almacenará en la base de datos.

Los informes estarán almacenados en la base de datos, XIndice, y se accederá a ellos mediante DOM o XPath.

3.2.2- Diagrama de flujo de la aplicación

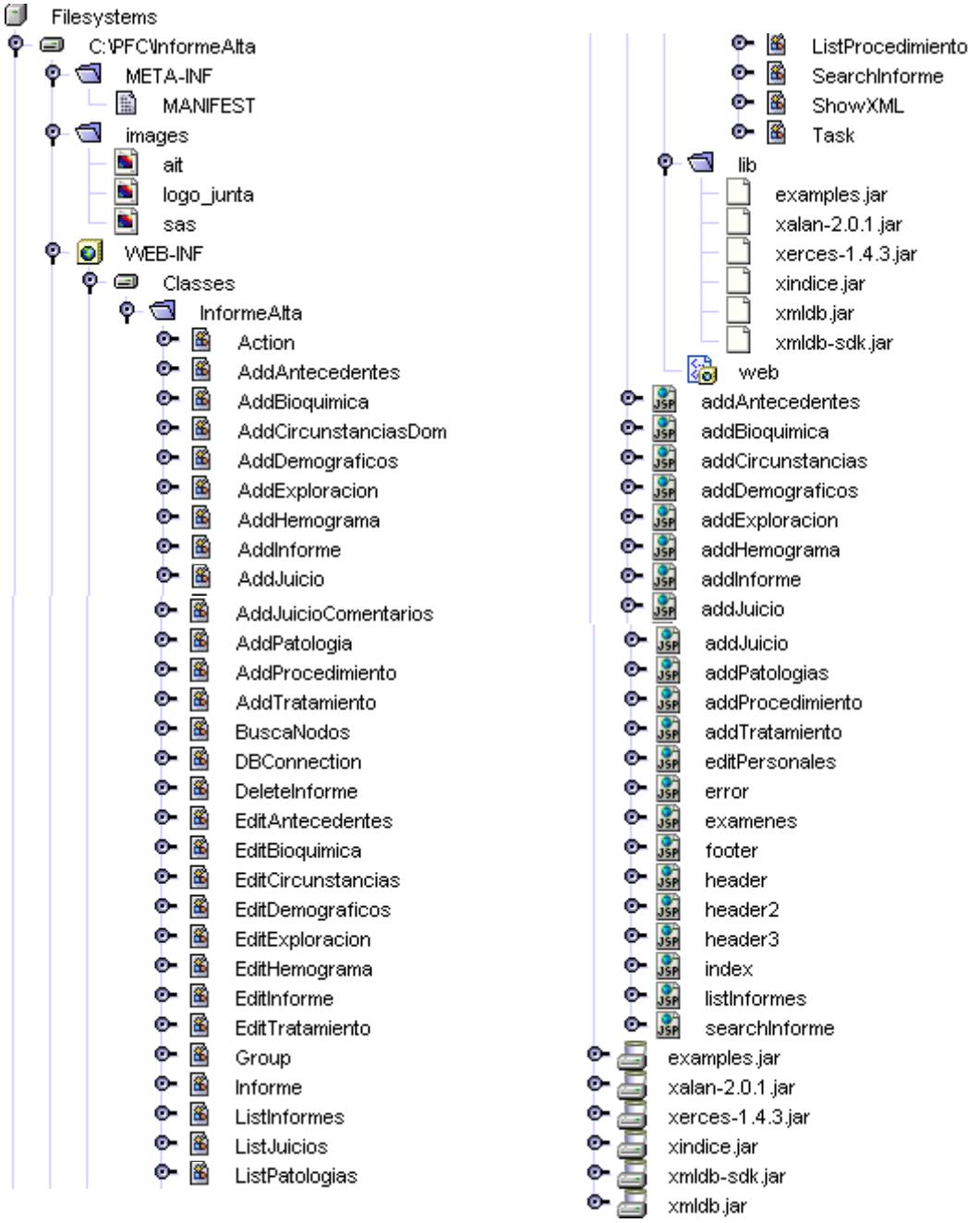
El diagrama de flujo nos permite conocer las alternativas que tiene el cliente (médico) para cumplimentar un Informe de Alta.

Podemos distinguir dos caminos, el primero en el que se procede de forma secuencial, y en el que se parte de los datos demográficos y conforme se van rellenando los formularios se va accediendo de forma automática al siguiente formulario. Y la segunda de ellas, en verde, en la que accediendo a los enlaces directos en la barra de navegación saltamos a los formularios que deseemos, de forma que ahorramos tiempo en el caso de que solo deseemos acceder a un formulario, y evitamos tener que pasar por los formulario anteriores al deseado.



3.2.3- Organización de los directorios

Podemos ver en este esquema la estructura definida en el código de la aplicación, en la que aparecen los nombres de todos los archivos creados.



La aplicación web está formada por los siguientes directorios:

- **/InformeAlta:** Directorio raíz de la aplicación web, en el cual se colocan todos los archivos HTML y JSP que utiliza la aplicación. Se pueden crear subdirectorios adicionales para mantener cualquier otro recurso de tipo estático que forme parte de la aplicación web y constituyan la parte de acceso público desde cualquier navegador.

- **/InformeAlta/META-INF:** Directorio que contiene el fichero de manifiesto para la aplicación.

- **/InformeAlta/WEB-INF:** Directorio que contiene todos los recursos relacionados con la aplicación web que no se han colocado en el directorio raíz y que no deben servirse al cliente. Esto es importante, ya que este directorio no forma parte del documento público, por lo que ninguno de los ficheros que contenga va a poder ser enviado directamente a través del servidor web.
  En este directorio se coloca el archivo web.xml, donde se establece la configuración de la aplicación web

- **/InformeAlta/WEB-INF/classes/InformeAlta:** Directorio que contiene todos los servlets y cualquier otra clase de utilidad o complementaria que se necesite para la ejecución de la aplicación web. Normalmente contiene solamente archivos .class.

- **/InformeAlta/WEB-INF/lib:** Directorio que contiene los archivos Java de los que depende la aplicación web. Por ejemplo, si la aplicación web necesita acceso a base de datos a través de JDBC, en este directorio es donde deben colocarse los ficheros JAR que contengan el driver JDBC que proporcione el acceso a la base de datos. Normalmente contiene solamente archivos .jar. Las librerías necesarias para esta aplicación son examples.jar, xalan-2.0.1.jar, xerces-1.4.3.jar, xindice.jar, xmldb.jar y xmldb-sdk.jar.

- **/InformeAlta/WEB-INF/images:** Directorio que contiene las imágenes que se usan para la presentación de los formularios.

- **/InformeAlta/WEB-INF/tlds:** Directorio que contiene los archivos TLD, descriptor de la librería de etiquetas, en el caso de que la aplicación web utilice cualquier librería de etiquetas, o acciones personalizadas.

En esta estructura, el cargador de clases consulta en primer lugar el directorio classes y posteriormente el directorio lib, de forma que en el desarrollo se pueden colocar clases de usuario en el directorio classes y las

clases ya probadas o clases de terceros en el directorio lib; de este modo el cargador de clases resolverá en primer lugar las clases con las que se encuentran en desarrollo, y si no es capaz de encontrarlas en el directorio classes las buscará en los archivos del directorio lib.

Una segunda parte muy importante de toda aplicación web es su descriptor de configuración. Este descriptor no es más que un archivo XML de nombre web.xml, localizado en el directorio WEB-INF, que contiene la descripción de la configuración correspondiente a la aplicación web. En el caso anterior el archivo web.xml estará situado en el directorio /InformeAlta/WEB-INF. La información que contiene este descriptor puede incluir los siguientes elementos:

- Parámetros de inicialización del ServletContext
- Configuración de la sesión
- Definiciones de Servlets/JavaServer Pages
- Mapeado de Servlets/JavaServer Pages
- Mapeado de tipos MIME
- Configuración de seguridad
- Páginas de error
- Páginas de bienvenida

3.2.4- Estructura general del código

La estructura del código la podríamos dividir en varios apartados, claramente diferenciadas:

- Por una parte tenemos el servlet que nos permite conectarnos a la base de datos, DBConnection. También destacaremos aquí el servlet Task, cuya función es esencial en el desarrollo de la aplicación. Este servlet será el encargado de administrar las tareas a realizar, los servlets o los jsp a ejecutar. Se visitará cada vez que se pulse un botón del formulario y contendrá las instrucciones a seguir.

- En segundo lugar tenemos el conjunto de servlets que permiten la creación de un nuevo archivo XML, que almacenará los datos de un

nuevo informe. El diseño se ha realizado de manera que el código XML referente al paciente es creado, en su "totalidad" (sólo faltarán algunos campos que se explicaran a continuación), en el momento en el que se introducen los primeros datos del paciente. El servlet en el que se crea todo el código es en Action, y el conjunto de servlets y clases que permiten este primer formulario son Group, Person, y AddInforme.

- En tercer lugar los servlets que usamos para presentar y modificar la mayoría de los formularios (todos cuyos datos fueron creados en Action). Para acceder a estos datos, se selecciona primero el paciente al que nos estamos refiriendo, se construye el árbol Dom del archivo, y se navega por él para recuperar los valores existentes o introducir los recogidos de los formularios.

- Y por último los servlets que permiten crear nuevo código XML (código XML no creado en el servlet Action), y leerlo y mostrarlo al usuario. Este es el caso de los formularios referentes a Patologías, Juicio Clínico y Procedimientos. Estos formularios tienen distinta funcionalidad que los anteriores, ya que permiten crear nuevas patologías, juicios o procedimientos, y cada vez que hagamos esto añadiremos nuevo código a nuestro esquema XML, y también muestran una lista de las patologías, juicios o procedimientos existentes. En el caso de querer introducir uno nuevo se optó por crear el código XML en cada nuevo elemento, de forma que el número de patologías, juicios o procedimientos es indeterminado y podemos introducir todos los que deseemos sin ningún incoveniente.

Código a destacar

Del código usado para acceder a la base de datos cabe destacar:

- En primer lugar se define una cadena a la que llamamos colstring y que contiene la dirección de la colección de la base de datos a la que vamos a acceder:

```
public static final String XINDICEURI="xindice:///db/";

public static final String COLLECTIONURI="xmldb:" + XINDICEURI;

String colstring = COLLECTIONURI + "InformesDeAlta";
```

- En segundo lugar registro la base de datos dbxml:

```
db = (Database)Class.forName("org.apache.xindice.client.xmldb.
DatabaseImpl").newInstance();

DatabaseManager.registerDatabase(db);
```

- Para por último conseguir una instancia colección a la que se le pasa el parámetro colstring definido con anterioridad:

```
collection = DatabaseManager.getCollection( colstring )
```

La estructura de los formularios y del código XML con el que se trabaja hacen que el código para algunos de los formularios sea muy parecido, aunque siempre se diferencian en los datos que los caracterizan. Esto permite que algunas líneas del código se repitan, y que la estructura de los servlets sean similares entre ellos.

Para poder acceder a los datos de los pacientes hay que tener en cuenta que esta información está almacenada en una base de datos, Xindice. Esta base de datos puede contener distintas colecciones, y estas a su vez tendrán documentos. Con el servlet Action accedemos a la colección InformesDeAlta, que es donde se almacenaran los documentos que vayamos creando con la aplicación. Por lo tanto aún nos queda por ver cómo determinar el documento a elegir entre los que componen la colección, y tratar con él. Para ello usaremos las siguientes instrucciones:

- col = getCollection(request,response); para conseguir una instancia collection

- String dockey = request.getParameter("DOCKEY"); toma del formulario el dockey del paciente con el que estamos trabajando

- XMLResource resource =(XMLResource) col.getResource(dockey); para recuperar un documento de una coleccion:

- Document doc = (Document) resource.getContentAsDOM(); lo convierte a objeto Dom

- BuscaNodos busca = new BuscaNodos(); para poder usar las funciones diseñadas en esta clase y que se explicarán a continuación

- busca.setDato1(doc,dockey,"FechaAdmision",0,Fecha); para introducir un dato recogido del formulario

- String GPT = busca.getDato1(doc,dockey,"ValorCantidad",29); para recuperar un dato y mostrarlo posteriormente en el formulario

- resource.setContentAsDOM(doc); y col.storeResource(resource); para almacenar el documento

Para poder trabajar con el árbol Dom con mayor simplicidad, hemos creado una clase BuscaNodos, en las que existen las funciones que se han usado en los servlets anteriores, que permiten almacenar datos en los nodos o recuperar sus valores. Para hacer todo esto posible podemos destacar el siguiente código:

- NodeList lista = doc.getElementsByTagName(nodo); creamos una lista de nodos Dom que tienen la misma etiqueta “nodo”

- eleNode = (Element)lista.item(i); seleccionamos el nodo que ocupa el lugar “i” de esa lista

- textNode = eleNode.getFirstChild();

- textNode.setNodeValue(valor); almacena “valor” en ese nodo

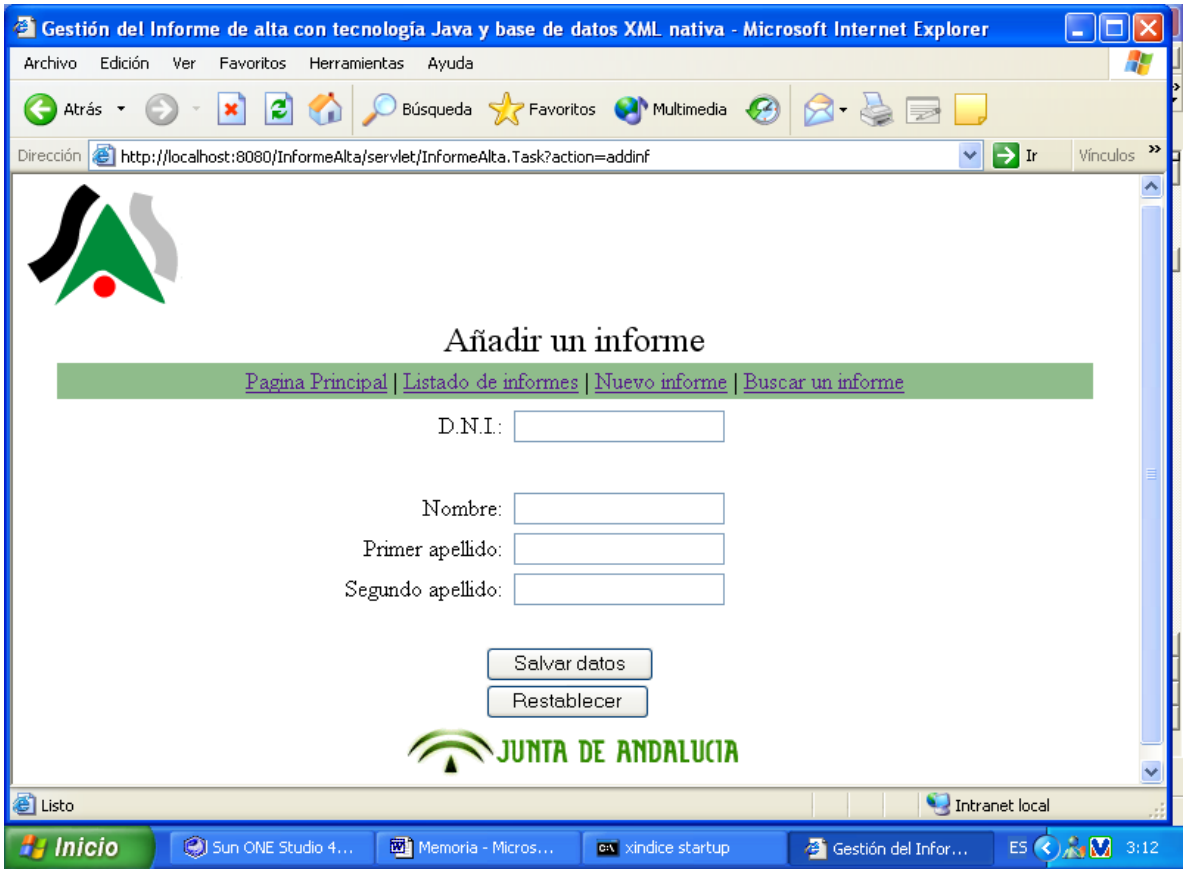
También es importante destacar el código usado en los formularios “especiales”, Patologías, Juicio clínico y Tratamiento, ya que en estos servlets crearemos el código XML que almacenará los datos recogidos del formulario.

Para crear estos nodos, lo primero que haré será colocarme en el nodo padre, y a partir de éste crear los nodos y asignarles las etiquetas, y los valores que sean necesarios:

- Element eleNode1 = doc.createElement ("Diagnostico"); creamos un nodo
- eleNode.appendChild (eleNode1); lo colocamos en su lugar dentro del árbol
- textNode3.appendChild(doc.createTextNode(FechaPatologia)); creamos un nodo texto con el valor contenido en FechaPatología

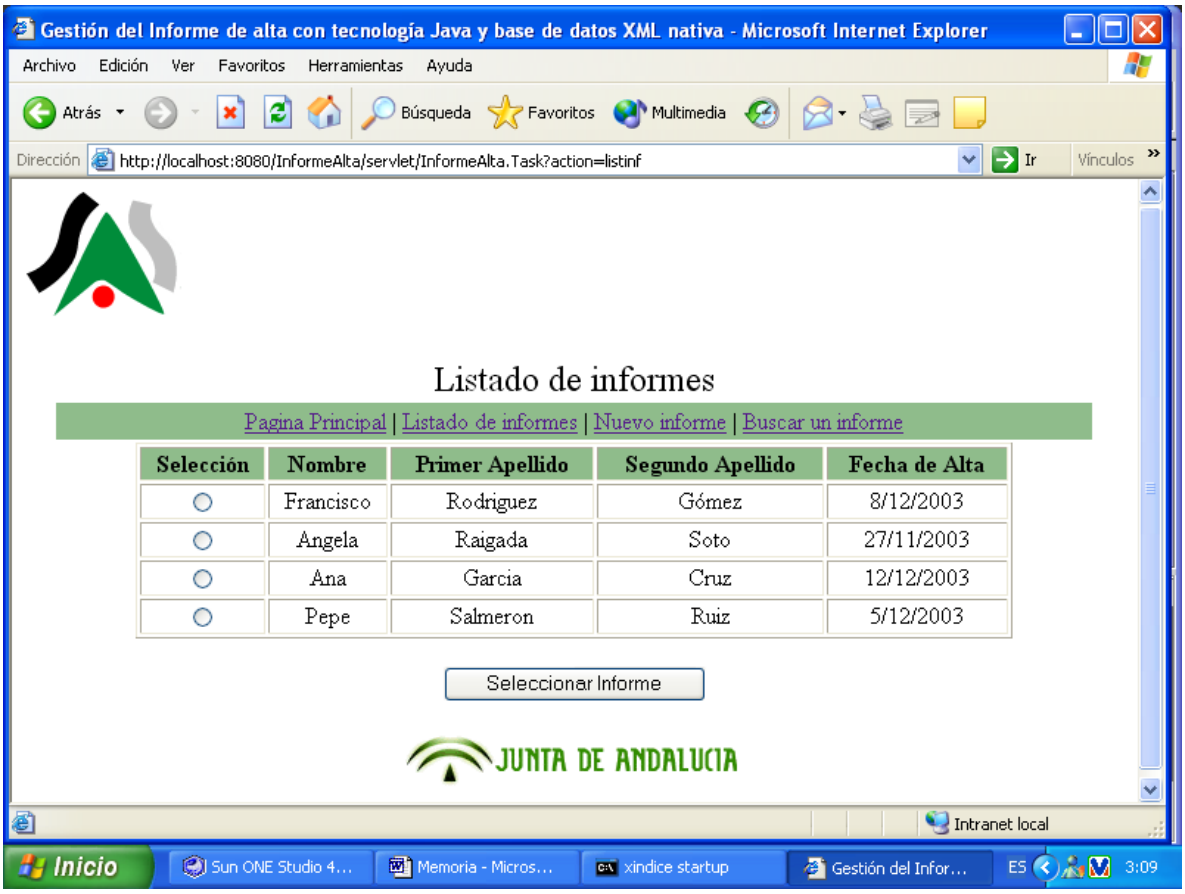
Nuevo Paciente

En el caso de la creación de un paciente, el proceso que seguiremos será el siguiente. El formulario addInforme.jsp nos permitirá introducir el DNI, nombre y dos apellidos del paciente, una vez introducidos, seleccionaremos el botón Añadir datos, con lo cual el servlet Task, indicará que las tareas a realizar son ejecutar el servlet AddInforme, y se creará el documento referente al paciente introducido.



El nombre de este documento vendrá determinado por un parámetro llamado dockey, que es un número aleatorio. Se ha elegido éste en lugar del DNI o del nombre o apellidos para evitar posibles confusiones en el caso de que hubiese varios informes de alta del mismo paciente, ya que en este caso con dockey no habría conflictos de ningún tipo.

Hemos definido que desde este formulario se pase al listado de pacientes, y ya desde aquí lo podremos seleccionar y pasar a rellenar el resto de formularios, o modificar alguno de los datos ya introducidos.



Una vez seleccionado el paciente de la lista, podremos modificar alguno de estos datos iniciales, en el caso de hacerlo volveremos a la lista de pacientes, si no es el caso seguiremos con la cadena de formularios.

Las barras de navegación, es decir, los posibles enlaces que permiten, serán diferentes según el formulario en el que nos encontremos, de forma que en estas pantallas iniciales definidas por addInforme.jsp y listInformes.jsp, sólo podremos volver a la Página principal, al Listado de Pacientes o Buscar un Paciente.

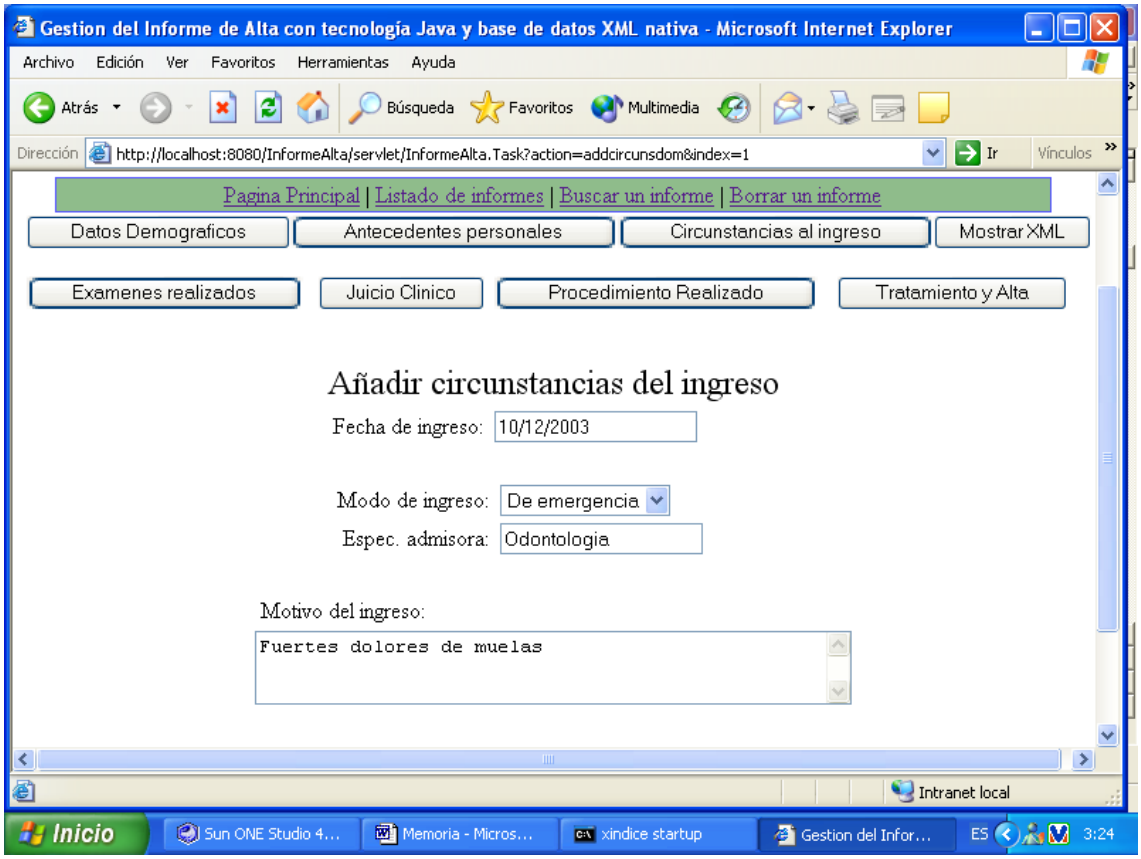
En este punto vamos a describir un poco la búsqueda de pacientes, se ha diseñado una búsqueda por atributos, DNI, nombre o apellidos. También se pensó el diseño de una búsqueda Xpath pero habría que indicarle todo el camino, y dadas las características del esquema no tiene mucho sentido, al igual que una búsqueda Dom, ya que tendríamos que pasarle al menos la etiqueta del parámetro, y no en todos los casos es inmediato.

Añadir/Modificar datos de un paciente(1)

En este apartado tendremos en cuenta a la mayoría de los formularios y dejaremos los tres casos especiales para un apartado posterior.

Aquí las barras de navegación permitirán más opciones, además de las dos anteriores de addInforme.jsp y listInformes.jsp, se permitirá acceder a cualquiera de los formularios, ya sea para rellenar los datos demográficos, antecedentes personales, circunstancias al ingreso…. También se podrá acceder a todos estos formularios de forma automática, al ir rellenando los formularios previos.

Para ver el funcionamiento con más detalle nos centraremos en uno de los formularios, por ejemplo, el de Circunstancias del ingreso.



Para llegar a este formulario tendremos las dos opciones comentadas anteriormente, la barra de navegación, o anotar los datos del formulario anterior. El servlet Task, recibirá action=addcircunsdom, y se procederá a ejecutar el servlet EditCircunstancias. En este servlet accederemos a la base de datos, en concreto al documento determinado por dockey, y tomaremos los valores de los parámetros correspondientes a este formulario, todo ello con Dom, siguiendo el código descrito en apartados anteriores.

Después de esto en Task se hará una llamada a la página Jsp addCircunstancias, en la que se le pasará el index del paciente con el que estamos, y accederemos al formulario y veremos los datos que para éste había en el documento, en el caso de que no estuviesen inicializados estarán a null.

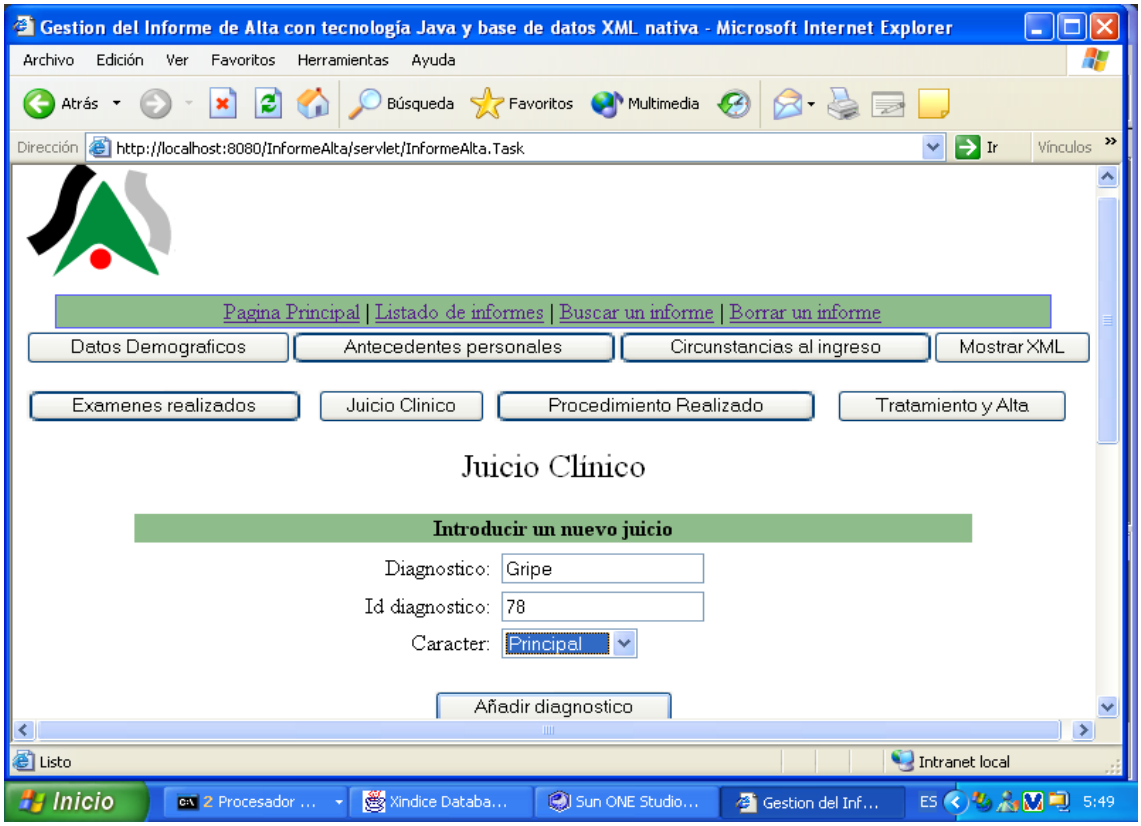
En este punto podremos modificar los datos que aparecen en el formulario o rellenarlo con nuevos datos, y una vez relleno, deberemos pulsar sobre el botón anotar datos, con lo que estaremos llamando al servlet Task pasándole action=addcircunstanciasdom, lo cual implica que se almacenarán los datos en Xindice, mediante el servlet AddCircunstanciasDom, y después se llamará a la página Jsp del siguiente formulario, que en este caso es examenes.jsp.

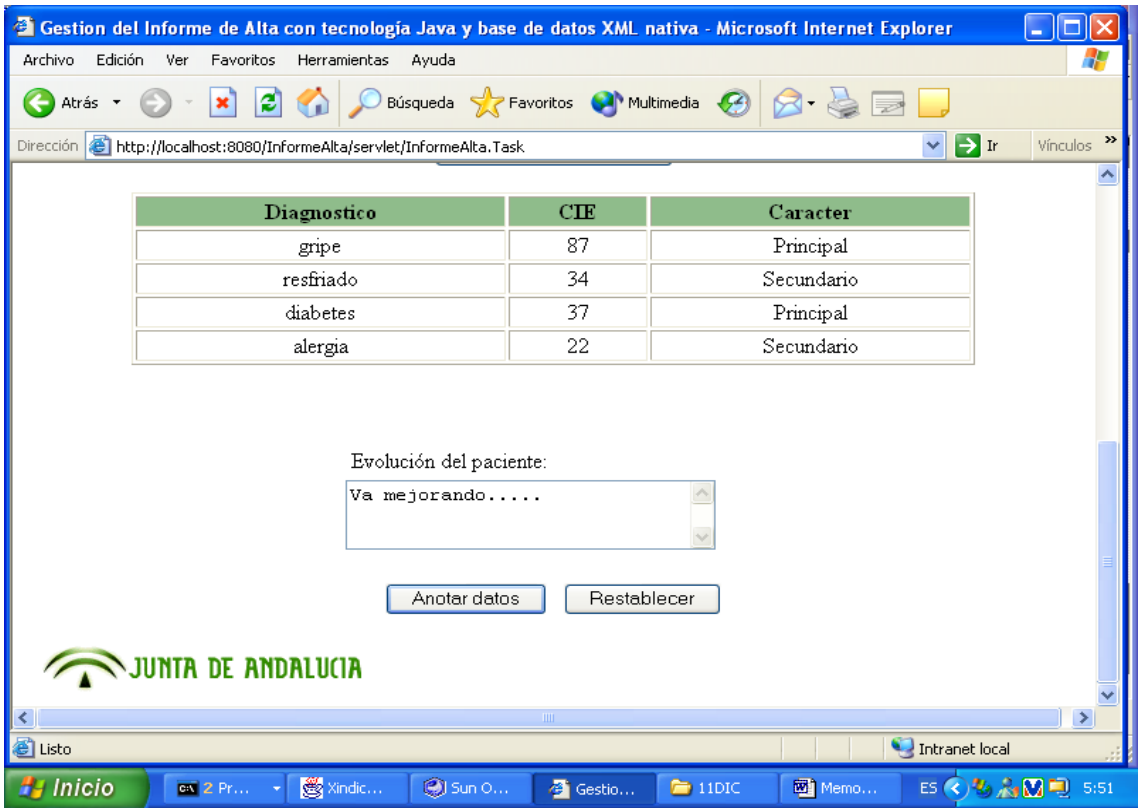
Añadir/Modificar datos de un paciente(2)

Aquí estudiaremos el caso de esos tres formularios que tienen una estructura distinta al resto. Esos tres formularios son, Patologías previas, Juicio Clínico y Procedimientos Realizados.

Estos formularios consistirán en lo siguiente, en primer lugar deben permitir añadir nuevos elementos, patologías, juicios o procedimientos, y en segundo lugar mostraran un listado con los elementos ya existente. La estructura con la que se han diseñado es muy similar, aunque cabe hacer una distinción con el formulario referente al Juicio clínico respecto a los otros dos, y es que en este caso además de las opciones de listado y añadir un nuevo elemento, se pueden rellenar varios campos adicionales.

Nos centraremos en el funcionamiento del formulario de Juicios clínicos ya que contiene los mismos puntos que los otros dos y alguno más.





La razón por la que el código de estos elementos no está en el documento XML que se creó al comienzo, es que si lo hubiésemos creado entonces, ahora sólo tendríamos un número de Juicios determinado, lo cual limitaría nuestro código. Por todo esto hemos preferido crear el código en este punto, es decir, una vez que hemos rellenado los campos del nuevo Juicio en el formulario y elegimos anotar datos, pasamos a crear toda la estructura de nodos y asignarles los valores.

También podríamos haber creado el código del resto de los formularios de forma dinámica, pero se prefirió hacerlo todo al principio porque el resto de ellos son estáticos, es decir, siempre son iguales, tienen el mismo número de elementos y sólo aparecen una vez.

Cuando decidimos entrar en este formulario, Juicio clínicos, por cualquiera de los dos caminos posibles, y como en todos los casos se siguen los pasos indicados en el servlet Task. En primer lugar se llamará al servlet que lee los datos del documento para poder hacer el listado, y después se accederá a la página addJuicios.jsp. Será desde esta página y una vez rellenados los campos indicados y pulsado, anotar datos, cuando se ejecute el servlet AddJuicios, en el que se crean los nodos que forman la estructura de los Juicios clínicos, se colocan en el nodo padre deseado, y se almacenan los datos introducidos.

4- LÍNEAS DE CONTINUACIÓN

4.1- LÍNEAS EN EL ÁMBITO DE LA APLICACIÓN

4.1.1- Mejora del Informe de Alta

La estructura XML elegida puede ampliarse y modificarse, para permitir almacenar un mayor número de datos del paciente. El formato elegido para esta aplicación ha sido en gran parte de otros formularios médicos. Sin embargo, los esquemas XML realizados son flexibles y permiten una mayor concreción de la información introducida o, simplemente, mayor cantidad de información.

Por otra parte en los formularios no se ha incluido validación de datos, que sería necesaria para evitar datos incorrectos y sin sentido.

El código XML está preparado para contemplar todas las concreciones y ampliaciones del Informe de Alta que se deseen hacer, sin más que modificar el código que se crea en el servlet Action o creando servlets que añadan ese código al ya existente.

4.1.2- Seguridad

Es evidente la necesidad de seguridad en el tipo de documentos con el que estamos trabajando, documentos que poseen un alto nivel de confidencialidad.

En este proyecto no se contempla nada en este respecto, pero se está realizando una colaboración con otro proyecto que permitiría una aplicación web segura, y que se integrará con el aquí realizado.

4.2- LÍNEAS EXTERNAS

4.2.1- Presentación de la información

Nuestro Proyecto termina una vez que se han cumplimentado los datos referentes al Informe de Alta del paciente.

No hemos centrado nuestro proyecto en el diseño de los formularios, por lo que se podrían modificar para darles una apariencia más estética. También sería posible crear el Informe de Alta a partir de los datos rellenados y presentar esta información con algunas de las herramientas basadas en XSL (*eXtensible Stylesheet Language*), que son las hojas de estilo propias de XML.

Otra posible representación del Informe de Alta podría ser en formato Word o con Adobe Acrobat.

4.2.2- Expansión de los esquemas XML

El Informe de Alta supone una porción muy pequeña del Historial Médico del paciente, es decir, los profesionales de la medicina trabajan con mucha más información referente al paciente.

Existen muchos más informes que se podrían añadir al creado como por ejemplo:

- Informe que incluya las acciones planificadas y la información relativa a la concertación de la cita y transporte del paciente.
- Informe de asistencias extrahospitalarias.
- Informe acerca de un paciente con el que no se ha tenido un contacto inmediato.
- Notificación de remisiones de un especialista a otro.
- Notificación de la admisión de un paciente en el hospital.
- Notificación de la muerte de un paciente en el hospital.

Por lo que se podrían añadir estos informes a la base de datos usada, Xindice, y se podrían crear nuevas colecciones para almacenar estos informes,

o crear documentos más grandes en los que se almacenasen varios de estos informes, de forma que no habría que rellenar algunos datos, como los demográficos en cada informe. Sin embargo Xindice no está indicada para documentos de gran tamaño, y habría que estudiar si permite esto o se sobrepasa su capacidad, o usar otra base de datos.

5- CONCLUSIONES

Es obvio que la proximidad del mundo de la tecnología con otras ciencias aparentemente nada relacionadas es cada vez mayor y mas intensa.

Como se ha comprobado durante la realización del Proyecto las tecnologías elegidas resultan muy útiles en el mundo laboral.

Además con la estructura seguida en la construcción de los servlets y jsp, la lectura y compresión del proyecto se ha conseguido que sea muy clara. La principal ventaja que esto proporciona es que añadir nueva información a los formularios o reutilizar el código de este proyecto es muy simple.

Los problemas encontrados en el desarrollo del proyecto han sido principalmente por la falta de conocimientos previos de las tecnologías a usar, lo que ha provocado que los avances fuesen lentos, aunque firmes.

La realización de este proyecto ha supuesto un éxito personal, ya que se ha conseguido adquirir una gran cantidad de conocimientos tanto teóricos como prácticos.

6- PLIEGO DE CONDICIONES

6.1- SERVLETS

Action.java

```
/*
 * Action.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.IOException;
import java.util.Hashtable;
import java.util.Vector;
import org.xmldb.api.DatabaseManager;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.apache.xpath.*;
import org.apache.xpath.objects.*;
import org.w3c.dom.*;
import javax.servlet.*;
import javax.servlet.http.*;
/**
 * @author  Irene Lavado Gomez
```

```
 */
/**
 * Definimos la clase Action, con la que crearemos el documento XML
 * en el que se almacenarán los datos recogidos de los formularios
 */
public abstract class Action extends HttpServlet {
    // Version del Api XMLDB que vamos a usar
    public final static String XMLDBAPIVERSION = "1.0";
     // Clase para mantener la conexion a la base de datos
    protected static DBConnection connection = new DBConnection();
     /**
     * Devuelve un objeto del tipo collection
     */
    public Collection getCollection(HttpServletRequest request,
HttpServletResponse response) throws javax.servlet.ServletException,
java.io.IOException {
        return connection.getCollectionInstance(request, response);
    }
     /**
     * Devuelve un vector de hashtables populated con todos los datos
     * almacenados en ResourceIterator
     */
    public Vector getContactVectorFromIterator(ResourceIterator results) {
         // una instancia de new Vector será devuelta
        Vector vector = new Vector();
        try {
            while (results.hasMoreResources()) {
                // Creo un XObject para usarlo en la búsqueda Xpath
```

```
XObject xo = new XObject();

// Creo un hastable para almacenarlo en el vector

Hashtable table = new Hashtable();

// Recupera el nodo siguiente

XMLResource resource = (XMLResource) results.nextResource();

Node originalnode = resource.getContentAsDOM();

// Consigue el parámetro key para este nodo

String dockey = resource.getDocumentId();

table.put("dockey", dockey);

// Usa parser Xalan xpath para extraer los datos

// almacenados en hastable

// DNI

xo = XPathAPI.eval(originalnode,
"/informe/DatosDemograficosPaciente/IdPaciente/ValorId");

table.put("ValorId", xo.toString());

 // Nombre y apellidos del paciente

xo = XPathAPI.eval(originalnode,
"/informe/DatosDemograficosPaciente/NombrePaciente/NombreEstructurado/N
ombre");

table.put("Nombre", xo.toString());

xo = XPathAPI.eval(originalnode,
"/informe/DatosDemograficosPaciente/NombrePaciente/NombreEstructurado/Pr
imerApellido");

table.put("PrimerApellido", xo.toString());

xo = XPathAPI.eval(originalnode,
"/informe/DatosDemograficosPaciente/NombrePaciente/NombreEstructurado/S
egundoApellido");

table.put("SegundoApellido", xo.toString());

// Fecha de alta

xo = XPathAPI.eval(originalnode,
"/informe/AdministracionAltaPaciente/DatosAlta/FechaAlta");
```

```
            table.put("FechaAlta", xo.toString());
            // Añade hashtable al vector to para pasarlo a la página JSP
            vector.add(table);
        }
      }
      catch (Exception e) {
         e.printStackTrace();
      }
      return vector;
   }
   /**
    * Creamos el documento XML a partir de los parámetros pasados,
    * el resto de los parámetros los inicializamos a null.
    */
   public String toXml(String DNI,String Nombre,String PrimerApellido,String SegundoApellido){
      String ourdoc;
      // En primer lugar inicializamos las variables no pasadas
      // Datos demograficos
      String ValorIdhc = null;
      String ValorIdss = null;
      String FechaNacimiento = null;
      String Sexo = null;
      String Calle = null;
      String Numero = null;
      String Piso = null;
      String Localidad = null;
      String Provincia = null;
```

```
String TipoDireccion = null;
String CodigoPostal = null;
String tfno1 = null;
String clase1 = null;
String tipo1 = null;
String tfno2 = null;
String clase2 = null;
String tipo2 = null;
String tfno3 = null;
String tipo3 = null;
String clase3 = null;
// Antecedentes personales
String AlergiaMedicamento = null;
String AlergiaOtros = null;
String Fumador = null;
String Bebedor = null;
String OtrosHabitos = null;
String Dialisis = null;
String ProcDialisis = null;
String Diabetes = null;
String HTA = null;
String Serologia = null;
String IngresoPrevio = null;
String ObservAntec = null;
String MedicHabitual = null;
String AntFamiliares = null;
// Patologías
```

```
String NombrePatologia = null;
String IdPatologia = null;
String FechaPatologia = null;
String CodClinPatologia = null;//este no se usa en formularios
// Juicio Clínico
String NombreDiagn = null;
String IdDiagn = null;
String Caracter = null;
String Evolucion = null;
String ObservJuicio = null;
String CodClinDiagn = null;//esto no se usa en formularios
// Procedimiento realizado
String NombreProced = null;
String IdProced = null;
String ObservProced = null;
String CodClinProced = null;//esta no se usa en formulario
// Primer examen a realizar: Exploración Física
String FechaExplor = null;
String Peso = null;
String FC = null;
String Temp = null;
String Talla = null;
String TA = null;
String AspectoGeneral = null;
String Cabeza = null;
String Torax = null;
String Abdomen = null;
```

```
String Extremidades = null;
String ObservExplor = null;
// Segundo examen a realizar: Hemograma
String FechaHemo = null;
String Hematies = null;
String Hemoglobina = null;
String Hematocrito = null;
String VCM = null;
String HCM = null;
String CHCM = null;
String RDW = null;
String Leucocitos = null;
String Segmentados = null;
String Linfocitos = null;
String Monocitos = null;
String Eosinofilos = null;
String Basofilos = null;
String Plaquetas = null;
String ObservHemo = null;
// Tercer examen a realizar: Prueba bioquímica
String FechaBioquim = null;
String Glucosa = null;
String Urea = null;
String Creatinina = null;
String AcidoUrico = null;
String Colesterol = null;
String Trigliceridos = null;
```

```
String CPK = null;
String ProteinaTotal = null;
String Albumina = null;
String GOT = null;
String GPT = null;
String GGT = null;
String BilirrubinaTotal = null;
String BilirrubinaDirecta = null;
String Fosfatasa = null;
String Lactico = null;
String Amilasa = null;
String Sodio = null;
String Potasio = null;
String Fosforo = null;
String Calcio = null;
String ObservBioquim = null;
// Añadir tratamiento
String FechaAlta = null;
String TipoAlta = null;
String LugarAlta = null;
String TratMedico = null;
String Administracion = null;
String ObservTrat = null;
// Motivo del ingreso
String FechaAdmision = null;
String ModoAdmision = null;
String EspecialidadAdmisora = null;
```

```
String MotivoIngreso = null;

// Otros

String Nombrerem = null;

String PrimerApellidorem = null;

String SegundoApellidorem = null;

String NombreOrganizacion = null;

String TipoOrganizacion = null;

String Nombrecont = null;

String PrimerApellidocont = null;

String SegundoApellidocont = null;

// Construimos el string XML

ourdoc = "<?xml version='1.0' encoding='ISO-8859-1' ?>";

ourdoc = "<informe>";

ourdoc = ourdoc + "<DatosDemograficosPaciente>";

ourdoc = ourdoc + "<IdPaciente>";

ourdoc = ourdoc + "<ValorId>" + DNI + "</ValorId>";

ourdoc = ourdoc + "<EsquemaId> DNI </EsquemaId>";

ourdoc = ourdoc + "</IdPaciente>";

ourdoc = ourdoc + "<IdPaciente>";

ourdoc = ourdoc + "<ValorId>" + ValorIdhc + "</ValorId>";

ourdoc = ourdoc + "<EsquemaId> Numero de la historia clinica
</EsquemaId>";

ourdoc = ourdoc + "</IdPaciente>";

ourdoc = ourdoc + "<IdPaciente>";

ourdoc = ourdoc + "<ValorId>" + ValorIdss + "</ValorId>";

ourdoc = ourdoc + "<EsquemaId> Numero de la Seguridad Social
</EsquemaId>";

ourdoc = ourdoc + "</IdPaciente>";
```

```
ourdoc = ourdoc + "<FechaNacimiento>" + FechaNacimiento + "</FechaNacimiento>";

ourdoc = ourdoc + "<Sexo>" + Sexo + "</Sexo>";

ourdoc = ourdoc + "<NombrePaciente>";

ourdoc = ourdoc + "<NombreEstructurado>";

ourdoc = ourdoc + "<Nombre>" + Nombre + "</Nombre>";

ourdoc = ourdoc + "<PrimerApellido>" + PrimerApellido + "</PrimerApellido>";

ourdoc = ourdoc + "<SegundoApellido>" + SegundoApellido + "</SegundoApellido>";

ourdoc = ourdoc + "</NombreEstructurado>";

ourdoc = ourdoc + "</NombrePaciente>";

ourdoc = ourdoc + "<DireccionPaciente>";

ourdoc = ourdoc + "<DireccionEstructurada>";

ourdoc = ourdoc + "<Calle>" + Calle + "</Calle>";

ourdoc = ourdoc + "<Numero>" + Numero + "</Numero>";

ourdoc = ourdoc + "<Piso>" + Piso + "</Piso>";

ourdoc = ourdoc + "<Localidad>" + Localidad + "</Localidad>";

ourdoc = ourdoc + "<Provincia>" + Provincia + "</Provincia>";

ourdoc = ourdoc + "</DireccionEstructurada>";

ourdoc = ourdoc + "<TipoDireccion>" + TipoDireccion + "</TipoDireccion>";

ourdoc = ourdoc + "<CodigoPostal>" + CodigoPostal + "</CodigoPostal>";

ourdoc = ourdoc + "</DireccionPaciente>";

ourdoc = ourdoc + "<TelecomPaciente>";

ourdoc = ourdoc + "<TelecomNoEstructurado>" + tfno1 +"</TelecomNoEstructurado>";

ourdoc = ourdoc + "<ClaseTelecom>" + clase1 + "</ClaseTelecom>";

ourdoc = ourdoc + "<TipoTelecom>" + tipo1 + "</TipoTelecom>";
```

```
ourdoc = ourdoc + "</TelecomPaciente>";

ourdoc = ourdoc + "<TelecomPaciente>";

ourdoc = ourdoc + "<TelecomNoEstructurado>" + tfno2 +"</TelecomNoEstructurado>";

ourdoc = ourdoc + "<ClaseTelecom>" + clase2 + "</ClaseTelecom>";

ourdoc = ourdoc + "<TipoTelecom>" + tipo2 + "</TipoTelecom>";

ourdoc = ourdoc + "</TelecomPaciente>";

ourdoc = ourdoc + "<TelecomPaciente>";

ourdoc = ourdoc + "<TelecomNoEstructurado>" + tfno3 +"</TelecomNoEstructurado>";

ourdoc = ourdoc + "<ClaseTelecom>" + clase3 + "</ClaseTelecom>";

ourdoc = ourdoc + "<TipoTelecom>" + tipo3 + "</TipoTelecom>";

ourdoc = ourdoc + "</TelecomPaciente>";

ourdoc = ourdoc + "</DatosDemograficosPaciente>";

ourdoc = ourdoc + "<AdministracionAltaPaciente>";

ourdoc = ourdoc + "<Remitente>";

ourdoc = ourdoc + "<McRemitente>";

ourdoc = ourdoc + "<AsNombre>";

ourdoc = ourdoc + "<NombreEstructurado>";

ourdoc = ourdoc + "<Nombre>" + Nombrerem + "</Nombre>";

ourdoc = ourdoc + "<PrimerApellido>" + PrimerApellidorem + "</PrimerApellido>";

ourdoc = ourdoc + "<SegundoApellido>" + SegundoApellidorem + "</SegundoApellido>";

ourdoc = ourdoc + "</NombreEstructurado>";

ourdoc = ourdoc + "</AsNombre>";

ourdoc = ourdoc + "</McRemitente>";

ourdoc = ourdoc + "</Remitente>";

ourdoc = ourdoc + "<DatosAdmision>";
```

```
ourdoc = ourdoc + "<Hospital>";

ourdoc = ourdoc + "<NombreOrganizacion>" + NombreOrganizacion + "</NombreOrganizacion>";

ourdoc = ourdoc + "<TipoOrganizacion>" + TipoOrganizacion + "</TipoOrganizacion>";

ourdoc = ourdoc + "</Hospital>";

ourdoc = ourdoc + "<EspecialidadAdmisora>" + EspecialidadAdmisora + "</EspecialidadAdmisora>";

ourdoc = ourdoc + "<MotivoIngreso>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica>" + MotivoIngreso + "</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</MotivoIngreso>";

ourdoc = ourdoc + "<FechaAdmision>" + FechaAdmision + "</FechaAdmision>";

ourdoc = ourdoc + "<ModoAdmision>" + ModoAdmision + "</ModoAdmision>";

ourdoc = ourdoc + "</DatosAdmision>";

ourdoc = ourdoc + "<DatosAlta>";

ourdoc = ourdoc + "<FechaAlta>" + FechaAlta + "</FechaAlta>";

ourdoc = ourdoc + "<TipoAlta>" + TipoAlta + "</TipoAlta>";

ourdoc = ourdoc + "<LugarAlta>" + LugarAlta + "</LugarAlta>";

ourdoc = ourdoc + "</DatosAlta>";

ourdoc = ourdoc + "<AsContacto>";

ourdoc = ourdoc + "<AsAltaFormaCortaProlongado>";

ourdoc = ourdoc + "<AsIdentificable>";

ourdoc = ourdoc + "<AsNombre>";

ourdoc = ourdoc + "<NombreEstructurado>";

ourdoc = ourdoc + "<Nombre>" + Nombrecont + "</Nombre>";

ourdoc = ourdoc + "<PrimerApellido>" + PrimerApellidocont + "</PrimerApellido>";
```

```
ourdoc = ourdoc + "<SegundoApellido>" + SegundoApellidocont + "</SegundoApellido>";

ourdoc = ourdoc + "</NombreEstructurado>";

ourdoc = ourdoc + "</AsNombre>";

ourdoc = ourdoc + "</AsIdentificable>";

ourdoc = ourdoc + "</AsAltaFormaCortaProlongado>";

ourdoc = ourdoc + "</AsContacto>";

ourdoc = ourdoc + "</AdministracionAltaPaciente>";

ourdoc = ourdoc + "<DatosClinicosAlta>";

ourdoc = ourdoc + "<DatosClinicosActualesAlta>";

ourdoc = ourdoc + "<Tratamiento>";

ourdoc = ourdoc + "<ProcedimientoRealizado>";

ourdoc = ourdoc + "<ProcedimientoRealizadoAlta>";

ourdoc = ourdoc + "</ProcedimientoRealizadoAlta>";

ourdoc = ourdoc + "</ProcedimientoRealizado>";

ourdoc = ourdoc + "</Tratamiento>";

ourdoc = ourdoc + "<Investigacion>";

ourdoc = ourdoc + "<TestRealizado>";

ourdoc = ourdoc + "<NombreTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Exploracion Fisica </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreTest>";

ourdoc = ourdoc + "</TestRealizado>";

ourdoc = ourdoc + "<FechaTest>" + FechaExplor + "</FechaTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Peso </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";
```

```
ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Peso + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> kg </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Hematocrito</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + FC + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> lpm </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Temperatura axilar </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";
```

```
ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Temp + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> C </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Talla </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Talla + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> cm </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Tension Arterial </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";
```

```
ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + TA + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> mmHg </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<LugarAnatomico>";

ourdoc = ourdoc + "<ParteCuerpo>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Aspecto General
</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</ParteCuerpo>";

ourdoc = ourdoc + "</LugarAnatomico>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<DescripcionMedida>" + AspectoGeneral +
"</DescripcionMedida>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<LugarAnatomico>";

ourdoc = ourdoc + "<ParteCuerpo>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Cabeza
</DescripcionCircunstanciaClinica>";
```

```
ourdoc = ourdoc + "</ParteCuerpo>";

ourdoc = ourdoc + "</LugarAnatomico>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<DescripcionMedida>" + Cabeza + "</DescripcionMedida>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<LugarAnatomico>";

ourdoc = ourdoc + "<ParteCuerpo>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Torax </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</ParteCuerpo>";

ourdoc = ourdoc + "</LugarAnatomico>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<DescripcionMedida>" + Torax + "</DescripcionMedida>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<LugarAnatomico>";

ourdoc = ourdoc + "<ParteCuerpo>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Abdomen </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</ParteCuerpo>";
```

```
ourdoc = ourdoc + "</LugarAnatomico>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<DescripcionMedida>" + Abdomen + "</DescripcionMedida>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<LugarAnatomico>";

ourdoc = ourdoc + "<ParteCuerpo>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Extremidades </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</ParteCuerpo>";

ourdoc = ourdoc + "</LugarAnatomico>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<DescripcionMedida>" + Extremidades + "</DescripcionMedida>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<Comentario>" + ObservExplor + "</Comentario>";

ourdoc = ourdoc + "</Investigacion>";

ourdoc = ourdoc + "<Investigacion>";

ourdoc = ourdoc + "<TestRealizado>";

ourdoc = ourdoc + "<NombreTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Hemograma </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreTest>";
```

```
ourdoc = ourdoc + "</TestRealizado>";

ourdoc = ourdoc + "<FechaTest>" + FechaHemo + "</FechaTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Hematies </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Hematies + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> x1000^2/uL </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 4.60 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 5.70 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";
```

```
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Hemoglobina </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Hemoglobina + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> g/dl </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 13.0 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 17.5 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Hematocrito </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
```

```
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + Hematocrito + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> % </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 40.0 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 54.0 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> VCM
</DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + VCM + "</ValorCantidad>";
```

```
ourdoc = ourdoc + "<UnidadCantidad> fl </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 82.0 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 98.0 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> HCM
</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + HCM + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> pg </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
```

```
ourdoc = ourdoc + "<ValorAltoRango> 27.0 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 31.0 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> CHCM </DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + CHCM + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> g/dl </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 31.0 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 38.0 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
```

```
ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> RDW </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + RDW + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> % </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 11.0 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 15.0 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Leucocitos </DescripcionCircunstanciaClinica>";
```

```
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + Leucocitos + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> x1000/uL </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 4.0 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 11.0 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Segmentados
</DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
```

```
ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Segmentados + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> % </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 42.0 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 75.0 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Linfocitos </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Linfocitos + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> % </UnidadCantidad>";
```

```
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 16.0 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 45.0 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Monocitos </DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + Monocitos + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> % </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 4.0 </ValorAltoRango>";
```

```
ourdoc = ourdoc + "<ValorBajoRango> 12.0 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Eosinofilos
</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Eosinofilos + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> % </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 1.0 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 4.0 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";
```

```
ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Basofilos </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Basofilos + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> % </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 0.0 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 2.0 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Plaquetas </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";
```

```
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + Plaquetas + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> % </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 140 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 150 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<Comentario>" + ObservHemo + "</Comentario>";
ourdoc = ourdoc + "</Investigacion>";
ourdoc = ourdoc + "<Investigacion>";
ourdoc = ourdoc + "<TestRealizado>";
ourdoc = ourdoc + "<NombreTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Prueba Bioquimica
</DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreTest>";
ourdoc = ourdoc + "</TestRealizado>";
ourdoc = ourdoc + "<FechaTest>" + FechaBioquim + "</FechaTest>";
```

```
ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Glucosa-s </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Glucosa + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> mg/dl </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 70 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 110 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Urea-s </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";
```

```
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + Urea + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> mg/dl </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 20 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 50 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Creatinina-s
</DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
```

```
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + Creatinina + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> mg/dl </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 0.70 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 1.50 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Acido urico-s
</DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + AcidoUrico + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> mg/dl </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
```

```
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 2.6 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 7.2 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Colesterol-s </DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + Colesterol + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> mg/dl </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 150 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 220 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
```

```
ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Trigliceridos-s
</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Trigliceridos + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> mg/dl </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 70 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 170 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";
```

```
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> CPK-s </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + CPK + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> U/L </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 10 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 195 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Proteina total-s </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
```

```
ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + ProteinaTotal + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> g/dl </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 6.0 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 8.0 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Albumina-s </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";
```

```
ourdoc = ourdoc + "<ValorCantidad>" + Albumina + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> g/dl </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 3.5 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 5.0 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> GOT(AST) </DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + GOT + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> U/L </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
```

```
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 0 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 37 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> GPT(ALT) </DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + GPT + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> U/L </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 0 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 40 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
```

```
ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> GGT </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + GGT + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> U/L </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 11 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 49 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";
```

```
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Bilirrubina total-s </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + BilirrubinaTotal + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> mg/dl </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 0.0 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 1.0 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Bilirrubina directa-s </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
```

```
ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + BilirrubinaDirecta + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> mg/dl </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 0.0 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 0.3 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Fosfotasa alcalina </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";
```

```
ourdoc = ourdoc + "<ValorCantidad>" + Fosfatasa + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> U/L </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 90 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 258 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Lactico deshidrogenasa </DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + Lactico + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> U/L </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
```

```
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 230 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 460 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Amilasa </DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + Amilasa + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> U/L </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 0 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 220 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
```

```
ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Sodio-s
</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Sodio + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> mEq/L </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 135 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 145 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";
```

```
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Potasio-s </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";

ourdoc = ourdoc + "</MuestraTestDetalles>";

ourdoc = ourdoc + "<MedidaTest>";

ourdoc = ourdoc + "<MedidaNumerica>";

ourdoc = ourdoc + "<Resultado>";

ourdoc = ourdoc + "<ValorCantidad>" + Potasio + "</ValorCantidad>";

ourdoc = ourdoc + "<UnidadCantidad> mEq/L </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 3.5 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 5.0 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<MuestraTest>";

ourdoc = ourdoc + "<NombreMuestraTest>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Fosforo-s </DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</NombreMuestraTest>";

ourdoc = ourdoc + "<MuestraTestDetalles>";

ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
```

```
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + Fosforo + "</ValorCantidad>";
ourdoc = ourdoc + "<UnidadCantidad> mg/dl </UnidadCantidad>";
ourdoc = ourdoc + "</Resultado>";
ourdoc = ourdoc + "</MedidaNumerica>";
ourdoc = ourdoc + "<LimiteReferencia>";
ourdoc = ourdoc + "<LimiteReferenciaNumerica>";
ourdoc = ourdoc + "<ValorAltoRango> 2.5 </ValorAltoRango>";
ourdoc = ourdoc + "<ValorBajoRango> 4.5 </ValorBajoRango>";
ourdoc = ourdoc + "</LimiteReferenciaNumerica>";
ourdoc = ourdoc + "</LimiteReferencia>";
ourdoc = ourdoc + "</MedidaTest>";
ourdoc = ourdoc + "</MuestraTest>";
ourdoc = ourdoc + "<MuestraTest>";
ourdoc = ourdoc + "<NombreMuestraTest>";
ourdoc = ourdoc + "<DescripcionCircunstanciaClinica> Calcio-s
</DescripcionCircunstanciaClinica>";
ourdoc = ourdoc + "</NombreMuestraTest>";
ourdoc = ourdoc + "<MuestraTestDetalles>";
ourdoc = ourdoc + "<TipoTejido> Sangre </TipoTejido>";
ourdoc = ourdoc + "</MuestraTestDetalles>";
ourdoc = ourdoc + "<MedidaTest>";
ourdoc = ourdoc + "<MedidaNumerica>";
ourdoc = ourdoc + "<Resultado>";
ourdoc = ourdoc + "<ValorCantidad>" + Calcio + "</ValorCantidad>";
```

```
ourdoc = ourdoc + "<UnidadCantidad> mg/dl </UnidadCantidad>";

ourdoc = ourdoc + "</Resultado>";

ourdoc = ourdoc + "</MedidaNumerica>";

ourdoc = ourdoc + "<LimiteReferencia>";

ourdoc = ourdoc + "<LimiteReferenciaNumerica>";

ourdoc = ourdoc + "<ValorAltoRango> 8.5 </ValorAltoRango>";

ourdoc = ourdoc + "<ValorBajoRango> 10.5 </ValorBajoRango>";

ourdoc = ourdoc + "</LimiteReferenciaNumerica>";

ourdoc = ourdoc + "</LimiteReferencia>";

ourdoc = ourdoc + "</MedidaTest>";

ourdoc = ourdoc + "</MuestraTest>";

ourdoc = ourdoc + "<Comentario>" + ObservBioquim + "</Comentario>";

ourdoc = ourdoc + "</Investigacion>";

ourdoc = ourdoc + "<MedicacionAlta>";

ourdoc = ourdoc + "<TipoMedicacionAlta>";

ourdoc = ourdoc + "<NombreMedicamento>";

ourdoc = ourdoc + "<DescripcionMedicamento>" + TratMedico + "</DescripcionMedicamento>";

ourdoc = ourdoc + "</NombreMedicamento>";

ourdoc = ourdoc + "<DosisNoEstructurada>" + Administracion + "</DosisNoEstructurada>";

ourdoc = ourdoc + "</TipoMedicacionAlta>";

ourdoc = ourdoc + "<Comentario>" + ObservTrat +"</Comentario>";

ourdoc = ourdoc + "</MedicacionAlta>";

ourdoc = ourdoc + "<Evolucion>" + Evolucion +"</Evolucion>";

ourdoc = ourdoc + "</DatosClinicosActualesAlta>";

ourdoc = ourdoc + "<Alertas>";

ourdoc = ourdoc + "<Alergia>";
```

```
ourdoc = ourdoc + "<AdvertenciaClinica>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica>"+AlergiaMedicamento
+"</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</AdvertenciaClinica>";

ourdoc = ourdoc + "<AdvertenciaClinica>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica>"+AlergiaOtros
+"</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</AdvertenciaClinica>";

ourdoc = ourdoc + "</Alergia>";

ourdoc = ourdoc + "<RiesgosModoVida>";

ourdoc = ourdoc + "<ConsumoAlcohol>";

ourdoc = ourdoc + "<AdvertenciaClinica>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica>"+Bebedor
+"</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</AdvertenciaClinica>";

ourdoc = ourdoc + "</ConsumoAlcohol>";

ourdoc = ourdoc + "<ConsumoTabaco>";

ourdoc = ourdoc + "<AdvertenciaClinica>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica>"+Fumador
+"</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</AdvertenciaClinica>";

ourdoc = ourdoc + "</ConsumoTabaco>";

ourdoc = ourdoc + "<AdvertenciaClinica>";

ourdoc = ourdoc + "<DescripcionCircunstanciaClinica>"+OtrosHabitos
+"</DescripcionCircunstanciaClinica>";

ourdoc = ourdoc + "</AdvertenciaClinica>";

ourdoc = ourdoc + "</RiesgosModoVida>";

ourdoc = ourdoc + "</Alertas>";

ourdoc = ourdoc + "<Antecedentes>";
```

```
ourdoc = ourdoc + "<HistoriaMedica>";

ourdoc = ourdoc + "<PatologiasPrevias>";

ourdoc = ourdoc + "</PatologiasPrevias>";

ourdoc = ourdoc + "</HistoriaMedica>";

ourdoc = ourdoc + "<AntecedentesFamiliares>";

ourdoc = ourdoc + "<Comentario>" + AntFamiliares + "</Comentario>";

ourdoc = ourdoc + "</AntecedentesFamiliares>";

ourdoc = ourdoc + "</Antecedentes>";

ourdoc = ourdoc + "<InformacionAdicionalPaciente>";

ourdoc = ourdoc + "<Necesidadespecial>";

ourdoc = ourdoc + "<AdvertenciaClinica>";

ourdoc = ourdoc + "<DescripcionCodigoClinico>" + Dialisis +
"</DescripcionCodigoClinico>";

ourdoc = ourdoc + "</AdvertenciaClinica>";

ourdoc = ourdoc + "<AdvertenciaClinica>";

ourdoc = ourdoc + "<DescripcionCodigoClinico>" + ProcDialisis +
"</DescripcionCodigoClinico>";

ourdoc = ourdoc + "</AdvertenciaClinica>";

ourdoc = ourdoc + "<AdvertenciaClinica>";

ourdoc = ourdoc + "<DescripcionCodigoClinico>" + Diabetes +
"</DescripcionCodigoClinico>";

ourdoc = ourdoc + "</AdvertenciaClinica>";

ourdoc = ourdoc + "<AdvertenciaClinica>";

ourdoc = ourdoc + "<DescripcionCodigoClinico>" + HTA +
"</DescripcionCodigoClinico>";

ourdoc = ourdoc + "</AdvertenciaClinica>";

ourdoc = ourdoc + "<AdvertenciaClinica>";

ourdoc = ourdoc + "<DescripcionCodigoClinico>" + Serologia +
"</DescripcionCodigoClinico>";
```

```
        ourdoc = ourdoc + "</AdvertenciaClinica>";
        ourdoc = ourdoc + "</Necesidadespecial>";
        ourdoc = ourdoc + "</InformacionAdicionalPaciente>";
        ourdoc = ourdoc + "</DatosClinicosAlta>";
        ourdoc = ourdoc + "</informe>";
        // Devuelve el string XML creado
        return ourdoc;
    }
}
```

AddAntecedentes.java

```
/*
 * AddAntecedentes.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
```

```
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase AddAntecedentes, esta clase sera la
 * encargada de recuperar los datos del formulario "Antecedentes
 * Familiares" y almacenarlos en el documento correspondiente al
 * paciente, todo ello mediante Dom
 */
public class AddAntecedentes extends Action {
   public boolean addAnt(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
       Collection col = null;
       try {
          // Consigue una instancia collecion
          col = getCollection(request,response);
          // Toma el dockey del paciente que estamos rellenando
          String dockey = request.getParameter("DOCKEY");
          // Tomo los parametros del formulario
          String Alergia = request.getParameter("ALERGIAMEDICAMENTO");
          String AlergiaOtros = request.getParameter("ALERGIAOTROS");
          String Fumador = request.getParameter("FUMADOR");
          String Bebedor = request.getParameter("BEBEDOR");
          String OtrosHabitos = request.getParameter("OTROSHABITOS");
          String Dialisis = request.getParameter("DIALISIS");
          String ProcDialisis = request.getParameter("PROCDIALISIS");
          String Diabetes = request.getParameter("DIABETES");
          String HTA = request.getParameter("HTA");
          String Serologia = request.getParameter("SEROLOGIA");
```

```
        String AntFamiliares = request.getParameter("ANTFAMILIARES");
       // Recuperar un documento de una coleccion
        XMLResource resource =(XMLResource) col.getResource(dockey);
        // Convierte a objeto DOM
        Document doc = (Document) resource.getContentAsDOM();
        // Almacenamos los valores tomados del formulario en
        // el documento XML correspondiente, usando las funciones
        // definidas en la clase BuscaNodos
        BuscaNodos busca = new BuscaNodos();
        busca.setDato2(doc,"AdvertenciaClinica",0,Alergia);
        busca.setDato2(doc,"AdvertenciaClinica",1,AlergiaOtros);
        busca.setDato2(doc,"AdvertenciaClinica",2,Bebedor);
        busca.setDato2(doc,"AdvertenciaClinica",3,Fumador);
        busca.setDato2(doc,"AdvertenciaClinica",4,OtrosHabitos);
        busca.setDato2(doc,"AdvertenciaClinica",5,Dialisis);
        busca.setDato2(doc,"AdvertenciaClinica",6,ProcDialisis);
        busca.setDato2(doc,"AdvertenciaClinica",7,Diabetes);
        busca.setDato2(doc,"AdvertenciaClinica",8,HTA);
        busca.setDato2(doc,"AdvertenciaClinica",9,Serologia);
        busca.setDato2(doc,"AntecedentesFamiliares",0,AntFamiliares);
        // Por último almacena el documento
        resource.setContentAsDOM(doc);
        col.storeResource(resource);
        }
      catch (IOException e) {
        System.out.println(e);
      }
```

```
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

AddBioquimica.java

```
/*
 * AddBioquimica.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
```

```
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase AddBioquimica, esta clase sera la
 * encargada de recuperar los datos del formulario "Pruebas
 * Bioquimicas" y almacenarlos en el documento correspondiente al
 * paciente, todo ello mediante Dom
 */
public class AddBioquimica extends Action {
    public boolean addBio(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
        Collection col= null;
        try{
            // Consigue una instancia collecion
            col = getCollection(request,response);
            // Toma el dockey del paciente que estamos rellenando
            String dockey = request.getParameter("DOCKEY");
            // Tomo los parametros del formulario
            String Fecha = request.getParameter("FECHABIOQUIM");
            String Glucosa = request.getParameter("GLUCOSA");
```

```
String Urea = request.getParameter("UREA");
String Creatinina = request.getParameter("CREATININA");
String AcidoUrico = request.getParameter("ACIDOURICO");
String Colesterol = request.getParameter("COLESTEROL");
String Trigliceridos = request.getParameter("TRIGLICERIDOS");
String CPK = request.getParameter("CPK");
String ProteinaTotal = request.getParameter("PROTEINATOTAL");
String Albumina = request.getParameter("ALBUMINA");
String GOT = request.getParameter("GOT");
String GPT = request.getParameter("GPT");
String GGT = request.getParameter("GGT");
String BilirrubinaTotal = request.getParameter("BILIRRUBINATOTAL");
String BilirrubinaDirecta = request.getParameter("BILIRRUBINADIRECTA");
String Fosfatasa = request.getParameter("FOSFATASA");
String Lactico = request.getParameter("LACTICO");
String Amilasa = request.getParameter("AMILASA");
String Sodio = request.getParameter("SODIO");
String Potasio = request.getParameter("POTASIO");
String Fosforo = request.getParameter("FOSFORO");
String Calcio = request.getParameter("CALCIO");
String Observ = request.getParameter("OBSERVBIOQUIM");
// Recuperar un documento de una coleccion
XMLResource resource =(XMLResource) col.getResource(dockey);
// Convierte a objeto DOM
Document doc = (Document) resource.getContentAsDOM();
// Almacenamos los valores tomados del formulario en
// el documento XML correspondiente, usando las funciones
```

```
// definidas en la clase BuscaNodos
BuscaNodos busca = new BuscaNodos();
busca.setDato1(doc,"FechaTest",2,Fecha);
busca.setDato1(doc,"ValorCantidad",19,Glucosa);
busca.setDato1(doc,"ValorCantidad",20,Urea);
busca.setDato1(doc,"ValorCantidad",21,Creatinina);
busca.setDato1(doc,"ValorCantidad",22,AcidoUrico);
busca.setDato1(doc,"ValorCantidad",23,Colesterol);
busca.setDato1(doc,"ValorCantidad",24,Trigliceridos);
busca.setDato1(doc,"ValorCantidad",25,CPK);
busca.setDato1(doc,"ValorCantidad",26,ProteinaTotal);
busca.setDato1(doc,"ValorCantidad",27,Albumina);
busca.setDato1(doc,"ValorCantidad",28,GOT);
busca.setDato1(doc,"ValorCantidad",29,GPT);
busca.setDato1(doc,"ValorCantidad",30,GGT);
busca.setDato1(doc,"ValorCantidad",31,BilirrubinaTotal);
busca.setDato1(doc,"ValorCantidad",32,BilirrubinaDirecta);
busca.setDato1(doc,"ValorCantidad",33,Fosfatasa);
busca.setDato1(doc,"ValorCantidad",34,Lactico);
busca.setDato1(doc,"ValorCantidad",35,Amilasa);
busca.setDato1(doc,"ValorCantidad",36,Sodio);
busca.setDato1(doc,"ValorCantidad",37,Potasio);
busca.setDato1(doc,"ValorCantidad",38,Fosforo);
busca.setDato1(doc,"ValorCantidad",39,Calcio);
// En el caso de la etiqueta comentario, tenemos que
// tener en cuenta, que depende del número de
// procedimientos que se hayan introducido, por lo
```

```
            // que tenemos que hayar dicho numero
            NodeList listaproc = doc.getElementsByTagName("Procedimiento");
            int numproc = listaproc.getLength();
            busca.setDato1(doc,"Comentario",numproc+2,Observ);
            // Por último almacena el documento
            resource.setContentAsDOM(doc);
            col.storeResource(resource);
        }
        catch (IOException e) {
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

AddCircunstanciasDom.java

```
/*
 * AddCircunstanciasDom.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase AddCircuntstanciasDom, esta clase sera la
 * encargada de recuperar los datos del formulario "Circunstancias
 * al ingreso" y almacenarlos en el documento correspondiente al
 * paciente, todo ello mediante Dom
 */
public class AddCircunstanciasDom extends Action {
```

```
   public boolean addCircuDom(HttpServletRequest request,
HttpServletResponse response ) throws ServletException, IOException {

      Collection col = null;

      try {

         // Consigue una instancia collecion

         col = getCollection(request,response);

         // Toma el dockey del paciente que estamos rellenando

         String dockey = request.getParameter("DOCKEY");

         // Tomo los parametros del formulario

         String Fecha = request.getParameter("FECHAADMISION");

         String Modo = request.getParameter("MODOADMISION");

         String Especialidad =
request.getParameter("ESPECIALIDADADMISORA");

         String Motivo = request.getParameter("MOTIVOINGRESO");

         // Recuperar un documento de una coleccion

         XMLResource resource =(XMLResource) col.getResource(dockey);

         // Convierte a objeto DOM

         Document doc = (Document) resource.getContentAsDOM();

         // Almacenamos los valores tomados del formulario en

         // el documento XML correspondiente, usando las funciones

         // definidas en la clase BuscaNodos

         BuscaNodos busca = new BuscaNodos();

         busca.setDato1(doc,"FechaAdmision",0,Fecha);

         busca.setDato1(doc,"ModoAdmision",0,Modo);

         busca.setDato1(doc,"EspecialidadAdmisora",0,Especialidad);

         busca.setDato2(doc,"MotivoIngreso",0,Motivo);

         // Tambien asigno la Fecha de Ingreso a la fecha de realización

         // de la Exploración Física
```

```
            busca.setDato1(doc,"FechaTest",0,Fecha);
            // Por último almacena el documento
            resource.setContentAsDOM(doc);
            col.storeResource(resource);
            }
        catch (IOException e) {
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

AddDemograficos.java

```
/*
```

```
 * AddDemograficos.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase AddDemograficos, esta clase sera la
 * encargada de recuperar los datos del formulario "Datos
 * Demograficos" y almacenarlos en el documento correspondiente
 * al paciente, todo ello mediante Dom
 */
public class AddDemograficos extends Action {
    public boolean addDemo(HttpServletRequest request, HttpServletResponse
response ) throws ServletException, IOException {
```

```
        Collection col = null;
        try {
            // Consigue una instancia collecion
            col = getCollection(request,response);
            // Toma el dockey del paciente que estamos rellenando
            String dockey = request.getParameter("DOCKEY");
            // Tomo los parametros del formulario
            String ValorId = request.getParameter("VALORID");
            String ValorIdhc = request.getParameter("VALORIDHC");
            String ValorIdss = request.getParameter("VALORIDSS");
            String FechaNacimiento =
request.getParameter("FECHANACIMIENTO");
            String Sexo = request.getParameter("SEXO");
            String Nombre = request.getParameter("NOMBRE");
            String PrimerApellido = request.getParameter("PRIMERAPELLIDO");
            String SegundoApellido =
request.getParameter("SEGUNDOAPELLIDO");
            String Calle = request.getParameter("CALLE");
            String Numero = request.getParameter("NUMERO");
            String Piso = request.getParameter("PISO");
            String Localidad = request.getParameter("LOCALIDAD");
            String Provincia = request.getParameter("PROVINCIA");
            String TipoDireccion = request.getParameter("TIPODIRECCION");
            String CodigoPostal = request.getParameter("CODIGOPOSTAL");
            String tfno1 = request.getParameter("TFNO1");
            String clase1 = request.getParameter("CLASE1");
            String tipo1 = request.getParameter("TIPO1");
            String tfno2 = request.getParameter("TFNO2");
```

```
String clase2 = request.getParameter("CLASE2");
String tipo2 = request.getParameter("TIPO2");
String tfno3 = request.getParameter("TFNO3");
String clase3 = request.getParameter("CLASE3");
String tipo3 = request.getParameter("TIPO3");
//Recuperar un documento de una coleccion
XMLResource resource =(XMLResource) col.getResource(dockey);
// Convierte a objeto DOM
Document doc = (Document) resource.getContentAsDOM();
// Almacenamos los valores tomados del formulario en
// el documento XML correspondiente, usando las funciones
// definidas en la clase BuscaNodos
BuscaNodos busca = new BuscaNodos();
busca.setDato1(doc,"ValorId",1,ValorIdhc);
busca.setDato1(doc,"ValorId",2,ValorIdss);
busca.setDato1(doc,"FechaNacimiento",0,FechaNacimiento);
busca.setDato1(doc,"Sexo",0,Sexo);
busca.setDato1(doc,"Calle",0,Calle);
busca.setDato1(doc,"Numero",0,Numero);
busca.setDato1(doc,"Piso",0,Piso);
busca.setDato1(doc,"Localidad",0,Localidad);
busca.setDato1(doc,"Provincia",0,Provincia);
busca.setDato1(doc,"TipoDireccion",0,TipoDireccion);
busca.setDato1(doc,"CodigoPostal",0,CodigoPostal);
busca.setDato1(doc,"TelecomNoEstructurado",0,tfno1);
busca.setDato1(doc,"ClaseTelecom",0,clase1);
busca.setDato1(doc,"TipoTelecom",0,tipo1);
```

```
            busca.setDato1(doc,"TelecomNoEstructurado",1,tfno2);
            busca.setDato1(doc,"ClaseTelecom",1,clase2);
            busca.setDato1(doc,"TipoTelecom",1,tipo2);
            busca.setDato1(doc,"TelecomNoEstructurado",2,tfno3);
            busca.setDato1(doc,"ClaseTelecom",2,clase3);
            busca.setDato1(doc,"TipoTelecom",2,tipo3);
            //Por último almacena el documento
            resource.setContentAsDOM(doc);
            col.storeResource(resource);
            }
        catch (IOException e) {
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
             if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
```

```
    }
}
```

AddExploracion.java

```
/*
 * AddExploracion.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase AddExploracion, esta clase sera la
 * encargada de recuperar los datos del formulario "Exploracion
 * Fisica" y almacenarlos en el documento correspondiente al
 * paciente, todo ello mediante Dom
```

```
 */

public class AddExploracion extends Action {

   public boolean addExp(HttpServletRequest request, HttpServletResponse
response ) throws ServletException, IOException {

      Collection col = null;

      try {

         // Consigue una instancia collecion

         col = getCollection(request,response);

         // Toma el dockey del paciente que estamos rellenando

         String dockey = request.getParameter("DOCKEY");

         // Tomo los parametros del formulario

         String index = request.getParameter("INDEX");

         String Fecha = request.getParameter("FECHAEXPLOR");

         String Peso = request.getParameter("PESO");

         String FC = request.getParameter("FC");

         String Temp = request.getParameter("TEMP");

         String Talla = request.getParameter("TALLA");

         String TA = request.getParameter("TA");

         String Aspecto = request.getParameter("ASPECTOGENERAL");

         String Torax = request.getParameter("TORAX");

         String Extremidades = request.getParameter("EXTREMIDADES");

         String Cabeza = request.getParameter("CABEZA");

         String Abdomen = request.getParameter("ABDOMEN");

         String Observ = request.getParameter("OBSERVEXPLOR");

         // Recuperar un documento de una coleccion

         XMLResource resource =(XMLResource) col.getResource(dockey);

         // Convierte a objeto DOM

         Document doc = (Document) resource.getContentAsDOM();
```

```
        // Almacenamos los valores tomados del formulario en
        // el documento XML correspondiente, usando las funciones
        // definidas en la clase BuscaNodos
        BuscaNodos busca = new BuscaNodos();
        busca.setDato1(doc,"FechaTest",0,Fecha);
        busca.setDato1(doc,"ValorCantidad",0,Peso);
        busca.setDato1(doc,"ValorCantidad",1,FC);
        busca.setDato1(doc,"ValorCantidad",2,Temp);
        busca.setDato1(doc,"ValorCantidad",3,Talla);
        busca.setDato1(doc,"ValorCantidad",4,TA);
        busca.setDato1(doc,"DescripcionMedida",0,Aspecto);
        busca.setDato1(doc,"DescripcionMedida",1,Cabeza);
        busca.setDato1(doc,"DescripcionMedida",2,Torax);
        busca.setDato1(doc,"DescripcionMedida",3,Abdomen);
        busca.setDato1(doc,"DescripcionMedida",4,Extremidades);
        // En el caso de la etiqueta comentario, tenemos que
        // tener en cuenta, que depende del número de
        // procedimientos que se hayan introducido, por lo
        // que tenemos que hayar dicho numero
        NodeList listaproc = doc.getElementsByTagName("Procedimiento");
        int numproc = listaproc.getLength();
        busca.setDato1(doc,"Comentario",numproc,Observ);
        // Por último almacena el documento
        resource.setContentAsDOM(doc);
        col.storeResource(resource);
        }
      catch (IOException e) {
```

```
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

AddHemograma.java

```
/*
 * AddHemograma.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
```

```
package InformeAlta;

import java.io.*;

import java.net.*;

import javax.servlet.*;

import javax.servlet.http.*;

import org.xmldb.api.base.*;

import org.xmldb.api.modules.*;

import org.xmldb.api.DatabaseManager;

import org.w3c.dom.*;

/**

 * @author  Irene Lavado Gomez

 */

/**

 * Definimos la clase AddHemograma, esta clase sera la

 * encargada de recuperar los datos del formulario "Hemograma"

 * y almacenarlos en el documento correspondiente al

 * paciente, todo ello mediante Dom

 */

public class AddHemograma extends Action {

   public boolean addHem(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {

       Collection col= null;

       try{

          // Consigue una instancia collecion

          col = getCollection(request,response);

          // Toma el dockey del paciente que estamos rellenando

          String dockey = request.getParameter("DOCKEY");

          // Tomo los parametros del formulario
```

```
String Fecha = request.getParameter("FECHAHEMO");
String Hematies = request.getParameter("HEMATIES");
String Hemoglobina = request.getParameter("HEMOGLOBINA");
String Hematocrito = request.getParameter("HEMATOCRITO");
String VCM = request.getParameter("VCM");
String HCM = request.getParameter("HCM");
String CHCM = request.getParameter("CHCM");
String RDW = request.getParameter("RDW");
String Leucocitos = request.getParameter("LEUCOCITOS");
String Segmentados = request.getParameter("SEGMENTADOS");
String Linfocitos = request.getParameter("LINFOCITOS");
String Monocitos = request.getParameter("MONOCITOS");
String Eosinofilos = request.getParameter("EOSINOFILOS");
String Basofilos = request.getParameter("BASOFILOS");
String Plaquetas = request.getParameter("PLAQUETAS");
String Observ = request.getParameter("OBSERVHEMO");
// Recuperar un documento de una coleccion
XMLResource resource =(XMLResource) col.getResource(dockey);
// Convierte a objeto DOM
Document doc = (Document) resource.getContentAsDOM();
// Almacenamos los valores tomados del formulario en
// el documento XML correspondiente, usando las funciones
// definidas en la clase BuscaNodos
BuscaNodos busca = new BuscaNodos();
busca.setDato1(doc,"FechaTest",1,Fecha);
busca.setDato1(doc,"ValorCantidad",5,Hematies);
busca.setDato1(doc,"ValorCantidad",6,Hemoglobina);
```

```
            busca.setDato1(doc,"ValorCantidad",7,Hematocrito);
            busca.setDato1(doc,"ValorCantidad",8,VCM);
            busca.setDato1(doc,"ValorCantidad",9,HCM);
            busca.setDato1(doc,"ValorCantidad",10,CHCM);
            busca.setDato1(doc,"ValorCantidad",11,RDW);
            busca.setDato1(doc,"ValorCantidad",12,Leucocitos);
            busca.setDato1(doc,"ValorCantidad",13,Segmentados);
            busca.setDato1(doc,"ValorCantidad",14,Linfocitos);
            busca.setDato1(doc,"ValorCantidad",15,Monocitos);
            busca.setDato1(doc,"ValorCantidad",16,Eosinofilos);
            busca.setDato1(doc,"ValorCantidad",17,Basofilos);
            busca.setDato1(doc,"ValorCantidad",18,Plaquetas);
            // En el caso de la etiqueta comentario, tenemos que
            // tener en cuenta, que depende del número de
            // procedimientos que se hayan introducido, por lo
            // que tenemos que hayar dicho numero
            NodeList listaproc = doc.getElementsByTagName("Procedimiento");
            int numproc = listaproc.getLength();
            busca.setDato1(doc,"Comentario",numproc+1,Observ);
            // Por último almacena el documento
            resource.setContentAsDOM(doc);
            col.storeResource(resource);
        }
        catch (IOException e) {
            System.out.println(e);
        }
        catch ( Exception e) {
```

```
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
             if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

AddInforme.java

```
/*
 * AddInforme.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import javax.servlet.*;
```

```
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import java.net.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase AddInforme, en la que recogeremos los datos del
 * formulario y haremos una llamada a la función creada en Action que
 * nos permitirá crear el documento XML y almacenarlo en la base de datos
 */
public class AddInforme extends Action {
  // Añade el documento a la base de datos usando los valores pasados via form-post
  public boolean add(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
     Collection col = null;
     String ourdoc = null;
     try {
       // Consigue una copia de la instancia collection
       col = getCollection(request,response);
       // Recupera los datos del formulario via post almacendolos como strings
       String dockey = request.getParameter("DOCKEY");
       String DNI = request.getParameter("VALORID");
       String Nombre = request.getParameter("NOMBRE");
       String PrimerApellido = request.getParameter("PRIMERAPELLIDO");
       String SegundoApellido = request.getParameter("SEGUNDOAPELLIDO");
```

```
        // Crea el string XML a partir de los valores recuperados
        ourdoc = toXml(DNI,Nombre,PrimerApellido,SegundoApellido);
        // Crea el XMLResource y almacena el documento,
        // como le paso "", en lugar de uno de los parametros,
        // el nombre del documento vendra dado por dockey
        XMLResource resource = (XMLResource) col.createResource( "",
"XMLResource" );
        resource.setContent(ourdoc);
        col.storeResource(resource);
      } catch ( Exception e) {
        e.printStackTrace();
         // Si la respuesta es committed no hay mucho más que se pueda hacer
         if (response.isCommitted())
            return true;
        // Toma la excepción y envia al usuario a la página de error
        if (e.getMessage() != null ) {
          response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
        }
        else {
          response.sendRedirect("/InformeAlta/error.jsp" );
        }
      }
      return true;
   }
}
```

AddJuicio.java

```
/*
```

```
 * AddJuicio.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase AddJuicio, esta clase sera la
 * encargada de recuperar los datos del formulario "Juicio
 * Clinico" referentes a un nuevo juicio, crear los nodos
 * correspondientes y almacena los valores recogidos, en los
 * nodos creados, todo ello mediante Dom
 */
public class AddJuicio extends Action {
    public boolean addJui(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
```

```
Collection col= null;
try{
    // Consigue una instancia collecion
    col = getCollection(request,response);
    // Toma el dockey del paciente que estamos rellenando
    String dockey = request.getParameter("DOCKEY");
    // Tomo los parametros del formulario
    String NombreDiagn = request.getParameter("NOMBREDIAGN");
    String IdDiagn = request.getParameter("IDDIAGN");
    String Caracter = request.getParameter("CARACTER");
    //Recuperar un documento de una coleccion
    XMLResource resource =(XMLResource) col.getResource(dockey);
    // Convierte a objeto DOM
    Document doc = (Document) resource.getContentAsDOM();
    // Ahora tengo que crear los nodos y asignarles los
    // valores que hemos tomado del formulario
    // Para crear los nodos tengo que localizar el nodo
    // sobre el que va a colgar
    Node eleNode0 = null;
    NodeList lista =
doc.getElementsByTagName("DatosClinicosActualesAlta");
    eleNode0 = (Element)lista.item(0);
    // Una vez localizado creo el nodo y luego lo coloco
    Element eleNode = doc.createElement ("Patologia");
    Element eleNode1 = doc.createElement ("DiagnosticoAlta");
    eleNode.appendChild (eleNode1);
    Element textNode3 = doc.createElement ("ImportanciaDiagnostico");
    textNode3.appendChild( doc.createTextNode(Caracter) );
```

```
        eleNode.appendChild (textNode3);
        Element eleNode2 = doc.createElement ("Diagnostico");
        eleNode1.appendChild (eleNode2);
        Element eleNode3 = doc.createElement ("NombrePatologia");
        eleNode2.appendChild (eleNode3);
        Element eleNode4 = doc.createElement ("CodigoClinico");
        eleNode3.appendChild (eleNode4);
        Element textNode2 = doc.createElement ("DescripcionCodigoClinico");
        textNode2.appendChild( doc.createTextNode(NombreDiagn) );
        eleNode3.appendChild (textNode2);
        Element eleNodev = doc.createElement ("ValorCodigoClinico");
        eleNodev.appendChild( doc.createTextNode("null") );
        eleNode4.appendChild (eleNodev);
        Element eleNode5 = doc.createElement ("EsquemaCodigoClinico");
        eleNode4.appendChild (eleNode5);
        Element textNode1 = doc.createElement ("IdEsquemaCodigoClinico");
        textNode1.appendChild( doc.createTextNode(IdDiagn) );
        eleNode5.appendChild (textNode1);
        eleNode0.appendChild (eleNode);
        // Por último almacena el documento
        resource.setContentAsDOM(doc);
        col.storeResource(resource);
        }
      catch (IOException e) {
        System.out.println(e);
      }
      catch ( Exception e) {
```

```
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
             if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

AddJuicioComentarios.java

```
/*
 * AddJuicioComentarios.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
```

```
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase AddJuicioComentarios, esta clase sera la
 * encargada de recuperar los comentarios del formulario "Juicio
 * Clinico" y almacenarlos en el documento correspondiente al
 * paciente, todo ello mediante Dom
 */
public class AddJuicioComentarios extends Action {
   public boolean addJuiCom(HttpServletRequest request,
HttpServletResponse response ) throws ServletException, IOException {
       Collection col= null;
       try{
          // Consigue una instancia collecion
          col = getCollection(request,response);
          // Toma el dockey del paciente que estamos rellenando
          String dockey = request.getParameter("DOCKEY");
          // Tomo los parametros del formulario
          String Evolucion = request.getParameter("EVOLUCION");
          String ObservJuicio = request.getParameter("OBSERVJUICIO");
          // Recuperar un documento de una coleccion
```

```
            XMLResource resource =(XMLResource) col.getResource(dockey);
            // Convierte a objeto DOM
            Document doc = (Document) resource.getContentAsDOM();
            // Almacenamos los valores tomados del formulario en
            // el documento XML correspondiente, usando las funciones
            // definidas en la clase BuscaNodos
            BuscaNodos busca = new BuscaNodos();
            busca.setDato1(doc,"Evolucion",0,Evolucion);
            // Por último almacena el documento
            resource.setContentAsDOM(doc);
            col.storeResource(resource);
            }
        catch (IOException e) {
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
```

```
        }
        return true;
    }
}
```

AddPatologia.java

```
/*
 * AddPatologia.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase AddPatoloia, esta clase sera la
 * encargada de recuperar los datos del formulario "Patologias
```

```
 * Previas" referentes a una nueva patologia, crear los nodos
 * correspondientes y almacena los valores recogidos, en los
 * nodos creados, todo ello mediante Dom
 */
public class AddPatologia extends Action {
    public boolean addPat(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
        Collection col = null;
        try {
            // Consigue una instancia collecion
            col = getCollection(request,response);
            // Toma el dockey del paciente que estamos rellenando
            String dockey = request.getParameter("DOCKEY");
            // Tomo los parametros del formulario
            String NombrePatologia = request.getParameter("NOMBREPATOLOGIA");
            String IdPatologia = request.getParameter("IDPATOLOGIA");
            String FechaPatologia = request.getParameter("FECHAPATOLOGIA");
            //Recuperar un documento de una coleccion
            XMLResource resource =(XMLResource) col.getResource(dockey);
            // Convierte a objeto DOM
            Document doc = (Document) resource.getContentAsDOM();
            // Ahora tengo que crear los nodos y asignarles los
            // valores que hemos tomado del formulario
            // Para crear los nodos tengo que localizar el nodo
            // sobre el que va a colgar
            Node eleNode0 = null;
            NodeList lista = doc.getElementsByTagName("PatologiasPrevias");
```

```
eleNode0 = (Element)lista.item(0);
// Una vez localizado creo el nodo y luego lo coloco
Element eleNode = doc.createElement ("RegistroPatologia");
Element eleNode1 = doc.createElement ("Diagnostico");
eleNode.appendChild (eleNode1);
Element textNode3 = doc.createElement ("FechaDiagnostico");
textNode3.appendChild( doc.createTextNode(FechaPatologia) );
eleNode.appendChild (textNode3);
Element eleNode2 = doc.createElement ("NombrePatologia");
eleNode1.appendChild (eleNode2);
Element eleNode3 = doc.createElement ("CodigoClinico");
eleNode2.appendChild (eleNode3);
Element textNode2 = doc.createElement ("DescripcionCodigoClinico");
textNode2.appendChild( doc.createTextNode(NombrePatologia) );
eleNode2.appendChild (textNode2);
Element eleNodev = doc.createElement ("ValorCodigoClinico");
eleNodev.appendChild( doc.createTextNode("null") );
eleNode3.appendChild (eleNodev);
Element eleNode4 = doc.createElement ("EsquemaCodigoClinico");
eleNode3.appendChild (eleNode4);
Element textNode1 = doc.createElement ("IdEsquemaCodigoClinico");
textNode1.appendChild( doc.createTextNode(IdPatologia) );
eleNode4.appendChild (textNode1);
eleNode0.appendChild (eleNode);
//Por último almacena el documento
resource.setContentAsDOM(doc);
col.storeResource(resource);
```

```
            }
        catch (IOException e) {
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

AddProcedimiento.java

```
/*
 * AddProcedimiento.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
```

```
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase AddProcedimiento, esta clase sera la
 * encargada de recuperar los datos del formulario "Procedimientos
 * Realizados" referentes a un nuevo procedimiento, crear los nodos
 * correspondientes y almacena los valores recogidos, en los
 * nodos creados, todo ello mediante Dom
 */
public class AddProcedimiento extends Action {
   public boolean addProc(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
       Collection col= null;
       try{
          // Consigue una instancia collecion
          col = getCollection(request,response);
```

```
        // Toma el dockey del paciente que estamos rellenando
        String dockey = request.getParameter("DOCKEY");
        // Tomo los parametros del formulario
        String NombreProced = request.getParameter("NOMBREPROCED");
        String IdProced = request.getParameter("IDPROCED");
        String ObservProced = request.getParameter("OBSERVPROCED");
        //Recuperar un documento de una coleccion
        XMLResource resource =(XMLResource) col.getResource(dockey);
        // Convierte a objeto DOM
        Document doc = (Document) resource.getContentAsDOM();
        // Ahora tengo que crear los nodos y asignarles los
        // valores que hemos tomado del formulario
        // Para crear los nodos tengo que localizar el nodo
        // sobre el que va a colgar
        Node eleNode0 = null;
        NodeList lista =
doc.getElementsByTagName("ProcedimientoRealizadoAlta");
        eleNode0 = (Element)lista.item(0);
        // Una vez localizado creo el nodo y luego lo coloco
        // primero le asignamos valor = null, y despues el valor
        // del formulario, en el caso de que lo haya
        Element eleNode = doc.createElement ("Procedimiento");
        Element eleNode1 = doc.createElement ("NombreProcedimiento");
        eleNode.appendChild (eleNode1);
        Element textNode3 = doc.createElement ("Comentario");
        textNode3.appendChild( doc.createTextNode(ObservProced) );
        eleNode.appendChild (textNode3);
        Element eleNode2 = doc.createElement ("CodigoClinico");
```

```
        eleNode1.appendChild (eleNode2);
        Element textNode2 = doc.createElement ("DescripcionCodigoClinico");
        textNode2.appendChild( doc.createTextNode(NombreProced) );
        eleNode1.appendChild (textNode2);
        Element eleNodev = doc.createElement ("ValorCodigoClinico");
        eleNodev.appendChild( doc.createTextNode("null") );
        eleNode2.appendChild (eleNodev);
        Element eleNode3 = doc.createElement ("EsquemaCodigoClinico");
        eleNode2.appendChild (eleNode3);
        Element textNode1 = doc.createElement ("IdEsquemaCodigoClinico");
        textNode1.appendChild( doc.createTextNode(IdProced) );
        eleNode3.appendChild (textNode1);
        eleNode0.appendChild (eleNode);
        // Por último almacena el documento
        resource.setContentAsDOM(doc);
        col.storeResource(resource);
        }
    catch (IOException e) {
        System.out.println(e);
    }
    catch ( Exception e) {
        e.printStackTrace();
        // Si la respuesta es committed no hay mucho más que se pueda hacer
        if (response.isCommitted())
            return true;
        // Toma la excepción y envia al usuario a la página de error
        if (e.getMessage() != null ) {
```

```
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

AddTratamiento.java

```
/*
 * AddTratamiento.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
```

```
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase AddTratmiento, esta clase sera la
 * encargada de recuperar los datos del formulario "Tratamiento
 * y alta" y almacenarlos en el documento correspondiente
 * al paciente, todo ello mediante Dom
 */
public class AddTratamiento extends Action {
  public boolean addTra(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
      Collection col= null;
      try{
         // Consigue una instancia collecion
         col = getCollection(request,response);
         // Toma el dockey del paciente que estamos rellenando
         String dockey = request.getParameter("DOCKEY");
         // Tomo los parametros del formulario
         String Fecha = request.getParameter("FECHAALTA");
         String Tipo = request.getParameter("TIPOALTA");
         String Lugar = request.getParameter("LUGARALTA");
         String Trat = request.getParameter("TRATMEDICO");
         String Adm = request.getParameter("ADMINISTRACION");
         String Observ = request.getParameter("OBSERVTRAT");
         //Recuperar un documento de una coleccion
         XMLResource resource =(XMLResource) col.getResource(dockey);
         // Convierte a objeto DOM
```

```
        Document doc = (Document) resource.getContentAsDOM();
        // Almacenamos los valores tomados del formulario en
        // el documento XML correspondiente, usando las funciones
        // definidas en la clase BuscaNodos
        BuscaNodos busca = new BuscaNodos();
        busca.setDato1(doc,"FechaAlta",0,Fecha);
        busca.setDato1(doc,"TipoAlta",0,Tipo);
        busca.setDato1(doc,"LugarAlta",0,Lugar);
        busca.setDato1(doc,"DescripcionMedicamento",0,Trat);
        busca.setDato1(doc,"DosisNoEstructurada",0,Adm);
        // En el caso de la etiqueta comentario, tenemos que tener
        // en cuenta, que depende del número de procedimientos que se
        // hayan introducido, por lo que tenemos que hayar dicho numero
        NodeList listaproc = doc.getElementsByTagName("Procedimiento");
        int numproc = listaproc.getLength();
        busca.setDato1(doc,"Comentario",3+numproc,Observ);
        //Por último almacena el documento
        resource.setContentAsDOM(doc);
        col.storeResource(resource);
        }
    catch (IOException e) {
        System.out.println(e);
    }
    catch ( Exception e) {
        e.printStackTrace();
        // Si la respuesta es committed no hay mucho más que se pueda hacer
        if (response.isCommitted())
```

```
            return true;
          // Toma la excepción y envia al usuario a la página de error
          if (e.getMessage() != null ) {
            response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
          }
          else {
            response.sendRedirect("/InformeAlta/error.jsp" );
          }
        }
        return true;
    }
}
```

BuscaNodos.java

```
/*
 * BuscaNodos.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
```

```
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase BuscaNodo, que nos ayudará a buscar los valores
 * de los nodos y a almacenar los valores de los formularios en los
 * nodos correspondientes.  Como parámetros comunes a todas serán doc,
 * que es el nodo Dom que representa el documento sobre el que vamos
 * a trabajar, el nodo, que sera el nombre de la etiqueta del nodo que
 * buscamos, i, que es el número de hijo, y valor para el caso en el
 * que tengamos que almacenar un valor
 */
public class BuscaNodos{
 //Almacena el valor tomado del formulario en el nodo especificado
    public void setDato1(Document doc,String nodo, int i,String valor){
        Node eleNode,eleNode1, textNode = null;
        NodeList lista = doc.getElementsByTagName(nodo);
        eleNode = (Element)lista.item(i);
        textNode = eleNode.getFirstChild();
        textNode.setNodeValue(valor);
   }
   //Toma el valor del nodo especificado
   public String getDato1(Document doc,String nodo, int i){
        String valor = null;
```

```
        Node eleNode,eleNode1, textNode = null;
        NodeList lista = doc.getElementsByTagName(nodo);
        eleNode = (Element)lista.item(i);
        textNode = eleNode.getFirstChild();
        valor = textNode.getNodeValue();
        return valor;
    }
    //Almacena el valor tomado del formulario en el nodo siguiente al especificado
    public void setDato2(Document doc,String nodo, int i,String valor){
        Node eleNode,eleNode1, textNode = null;
        NodeList lista = doc.getElementsByTagName(nodo);
        eleNode = (Element)lista.item(i);
        eleNode1 = eleNode.getFirstChild();
        textNode = eleNode1.getFirstChild();
        textNode.setNodeValue(valor);
    }
   //Toma el valor del nodo siquiente al especificado
    public String getDato2(Document doc,String nodo, int i){
        String valor = null;
        Node eleNode,eleNode1, textNode = null;
        NodeList lista = doc.getElementsByTagName(nodo);
        eleNode = (Element)lista.item(i);
        eleNode1 = eleNode.getFirstChild();
        textNode = eleNode1.getFirstChild();
        valor = textNode.getNodeValue();
        return valor;
    }
```

```
}
```

DBConnection.java

```
/*
 * DBConnection.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
 import org.xmldb.api.base.*;
import org.apache.xindice.client.xmldb.*;
import org.xmldb.api.DatabaseManager;
import javax.servlet.*;
import javax.servlet.http.*;
import java.net.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase DBConnection, esta clase será la encargada
 * de conectarnos a la base de datos Xindice, y a la colección
 * que hayamos definido para almacenar nuestros pacientes
 */
public class DBConnection extends HttpServlet {
   // Definimos el path, en colstring, de la colección que vamos a usar
   public static final String XINDICEURI="xindice:///db/";
```

```
  public static final String COLLECTIONURI="xmldb:" + XINDICEURI;

  String colstring = COLLECTIONURI + "InformesDeAlta";

  protected static DBConnection connection = new DBConnection();

  protected static Database db = null;

  protected static Collection collection = null;

  public Collection getCollectionInstance(HttpServletRequest request,
HttpServletResponse response ) throws javax.servlet.ServletException,
java.io.IOException {

    try {

      if ( db == null ) {

        // Registra la base de datos dbxml handler

        db =
(Database)Class.forName("org.apache.xindice.client.xmldb.DatabaseImpl").ne
wInstance();

        DatabaseManager.registerDatabase(db);

        // Consigue una instancia collection

        collection = DatabaseManager.getCollection( colstring );

      }

    } catch ( Exception e) {

      e.printStackTrace();

       // Si la respuesta es committed no hay mucho más que se pueda hacer

       if (response.isCommitted())

          return collection;

      // Toma la excepción y envia al usuario a la página de error

      if (e.getMessage() != null ) {

        response.sendRedirect("/InformeAlta/error.jsp?error=" +
URLEncoder.encode(e.getMessage()) );

      }

      else {
```

```
            response.sendRedirect("/InformeAlta/error.jsp" );
          }
        }
        return collection;
    }
}
```

DeleteInforme.java

```
/*
 * DeleteInforme.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import java.net.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase DeleteInforme, esta clase será la encargada
```

```
 * de borrar los documentos almacenados en la base de datos. Para seleccionar

 * el documento a borrar usaremos el parámetro index

 */

public class DeleteInforme extends Action {

  public boolean delete(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {

    Collection col = null;

    try {

      // Consigue una referencia de la sesión

      HttpSession session = request.getSession(true);

      // Consigue una instancia collecion

      col = getCollection(request,response);

      // Consigue el index que vamos a borrar

      int index = Integer.parseInt((String)request.getParameter("index"));

      // Pone el Group fuera de la sesión, y consigue a la instancia

      // informe referente a ese index

      Group group = (Group)session.getAttribute("group");

      Informe informe = (Informe)group.getInformeAt(index);

      // Toma el dockey para ese informe particular

      String dockey = informe.getDocKey();

      // Borra el documento

      col.removeResource(col.getResource(dockey) );

    } catch (Exception e) {

      e.printStackTrace();

       // Si la respuesta es committed no hay mucho más que se pueda hacer

        if (response.isCommitted())

            return true;
```

```
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

EditAntecedentes.java

```
/*
 * EditAntecedentes.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
```

```
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase EditAntecedentes, esta clase sera la encargada
 * de recuperar los datos del documento XML almacenado y mandarlos a
 * la página JSP correspondiente al formulario "Antecedentes
 * personales", todo ello mediante Dom
 */
public class EditAntecedentes extends Action {
   public boolean editAnt(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
       Collection col= null;
       try{
          // Consigue una instancia collection
          col = getCollection(request,response);
          // Recupera el dockey del paciente que estamos rellenando
          String dockey = request.getParameter("DOCKEY");
          // Recuperar un documento de una coleccion:
          XMLResource resource =(XMLResource) col.getResource(dockey);
          // Convierte a objeto Dom
          Document doc = (Document) resource.getContentAsDOM();
          // Leo los datos del documento y los almacena en strings para
          // después enviarlos a la página JSP, para ello usamos las
          // funciones definidas en la clase BuscaNodos
          BuscaNodos busca = new BuscaNodos();
```

```
            String Alergia = busca.getDato2(doc,"AdvertenciaClinica",0);
            String AlergiaOtros = busca.getDato2(doc,"AdvertenciaClinica",1);
            String Bebedor = busca.getDato2(doc,"AdvertenciaClinica",2);
            String Fumador = busca.getDato2(doc,"AdvertenciaClinica",3);
            String OtrosHabitos = busca.getDato2(doc,"AdvertenciaClinica",4);
            String Dialisis = busca.getDato2(doc,"AdvertenciaClinica",5);
            String ProcDialisis = busca.getDato2(doc,"AdvertenciaClinica",6);
            String Diabetes = busca.getDato2(doc,"AdvertenciaClinica",7);
            String HTA = busca.getDato2(doc,"AdvertenciaClinica",8);
            String Serologia = busca.getDato2(doc,"AdvertenciaClinica",9);
            String AntFamiliares =
busca.getDato2(doc,"AntecedentesFamiliares",0);
            request.setAttribute("Alergia",Alergia);
            request.setAttribute("AlergiaOtros",AlergiaOtros);
            request.setAttribute("Bebedor",Bebedor);
            request.setAttribute("Fumador",Fumador);
            request.setAttribute("OtrosHabitos",OtrosHabitos);
            request.setAttribute("Dialisis",Dialisis);
            request.setAttribute("ProcDialisis",ProcDialisis);
            request.setAttribute("Diabetes",Diabetes);
            request.setAttribute("HTA",HTA);
            request.setAttribute("Serologia",Serologia);
            request.setAttribute("AntFamiliares",AntFamiliares);
            // Almacenar el documento
            resource.setContentAsDOM(doc);
            col.storeResource(resource);
        }
        catch (IOException e) {
```

```
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
           // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

EditBioquimica.java

```
/*
 * EditBioquimica.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
```

```
package InformeAlta;

import java.io.*;

import java.net.*;

import javax.servlet.*;

import javax.servlet.http.*;

import org.xmldb.api.base.*;

import org.xmldb.api.modules.*;

import org.xmldb.api.DatabaseManager;

import org.w3c.dom.*;

/**

 * @author  Irene Lavado Gomez

 */

/**

 * Definimos la clase EditBioquimica, esta clase sera la encargada

 * de recuperar los datos del documento XML almacenado y mandarlos a

 * la página JSP correspondiente al formulario "Prueba Bioquimica",

 * todo ello mediante Dom

 */

public class EditBioquimica extends Action {

    public boolean editBio(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {

        Collection col= null;

        try{

            // Consigue una instancia collection

            col = getCollection(request,response);

            // Recupera el dockey del paciente que estamos rellenando

            String dockey = request.getParameter("DOCKEY");

            // Recuperar un documento de una coleccion:
```

```
XMLResource resource =(XMLResource) col.getResource(dockey);
// Convierte a objeto Dom
Document doc = (Document) resource.getContentAsDOM();
// Leo los datos del documento y los almacena en strings para
// después enviarlos a la página JSP, para ello usamos las
// funciones definidas en la clase BuscaNodos
BuscaNodos busca = new BuscaNodos();
String Fecha = busca.getDato1(doc,"FechaTest",2);
String Glucosa = busca.getDato1(doc,"ValorCantidad",19);
String Urea = busca.getDato1(doc,"ValorCantidad",20);
String Creatinina = busca.getDato1(doc,"ValorCantidad",21);
String AcidoUrico = busca.getDato1(doc,"ValorCantidad",22);
String Colesterol = busca.getDato1(doc,"ValorCantidad",23);
String Trigliceridos = busca.getDato1(doc,"ValorCantidad",24);
String CPK = busca.getDato1(doc,"ValorCantidad",25);
String ProteinaTotal = busca.getDato1(doc,"ValorCantidad",26);
String Albumina = busca.getDato1(doc,"ValorCantidad",27);
String GOT = busca.getDato1(doc,"ValorCantidad",28);
String GPT = busca.getDato1(doc,"ValorCantidad",29);
String GGT = busca.getDato1(doc,"ValorCantidad",30);
String BilirrubinaTotal = busca.getDato1(doc,"ValorCantidad",31);
String BilirrubinaDirecta = busca.getDato1(doc,"ValorCantidad",32);
String Fosfatasa = busca.getDato1(doc,"ValorCantidad",33);
String Lactico = busca.getDato1(doc,"ValorCantidad",34);
String Amilasa = busca.getDato1(doc,"ValorCantidad",35);
String Sodio = busca.getDato1(doc,"ValorCantidad",36);
String Potasio = busca.getDato1(doc,"ValorCantidad",37);
```

```
String Fosforo = busca.getDato1(doc,"ValorCantidad",38);
String Calcio = busca.getDato1(doc,"ValorCantidad",39);
// En el caso de la etiqueta comentario, tenemos que tener
// en cuenta, que depende del número de procedimientos que se
// hayan introducido, por lo que tenemos que hayar dicho numero
NodeList listaproc = doc.getElementsByTagName("Procedimiento");
int numproc = listaproc.getLength();
String Observ = busca.getDato1(doc,"Comentario",2);
request.setAttribute("Glucosa",Glucosa);
request.setAttribute("Urea",Urea);
request.setAttribute("Creatinina",Creatinina);
request.setAttribute("AcidoUrico",AcidoUrico);
request.setAttribute("Colesterol",Colesterol);
request.setAttribute("CPK",CPK);
request.setAttribute("ProteinaTotal",ProteinaTotal);
request.setAttribute("Albumina",Albumina);
request.setAttribute("GOT",GOT);
request.setAttribute("GPT",GPT);
request.setAttribute("GGT",GGT);
request.setAttribute("BilirrubinaTotal",BilirrubinaTotal);
request.setAttribute("BilirruninaDirecta",BilirrubinaDirecta);
request.setAttribute("Fosfatasa",Fosfatasa);
request.setAttribute("Lactico",Lactico);
request.setAttribute("Amilasa",Amilasa);
request.setAttribute("Sodio",Sodio);
request.setAttribute("Potasio",Potasio);
request.setAttribute("Fosforo",Fosforo);
```

```
            request.setAttribute("Calcio",Calcio);
            request.setAttribute("Observ",Observ);
            // Almacenar el documento
            resource.setContentAsDOM(doc);
            col.storeResource(resource);
        }
        catch (IOException e) {
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

EditCircunstancias.java

```
/*
 * EditCircunstancias.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase EditCircunstancias, esta clase sera la encargada
 * de recuperar los datos del documento XML almacenado y mandarlos a
 * la página JSP correspondiente al formulario "Circunstancias al
 * ingreso", todo ello mediante Dom
 */
public class EditCircunstancias extends Action {
```

```
   public boolean editCircu(HttpServletRequest request, HttpServletResponse
response ) throws ServletException, IOException {
      Collection col = null;
      try {
         // Consigue una instancia collection
         col = getCollection(request,response);
         // Recupera el dockey del paciente que estamos rellenando
         String dockey = request.getParameter("DOCKEY");
         // Recuperar un documento de una coleccion:
         XMLResource resource =(XMLResource) col.getResource(dockey);
         // Convierte a objeto Dom
         Document doc = (Document) resource.getContentAsDOM();
         // Leo los datos del documento y los almacena en strings
         // para después enviarlos a la página JSP, para ello
         // usamos las funciones definidas en la clase BuscaNodos
         BuscaNodos busca = new BuscaNodos();
         String Fecha = busca.getDato1(doc,"FechaAdmision",0);
         String Modo = busca.getDato1(doc,"ModoAdmision",0);
         String Especialidad = busca.getDato1(doc,"EspecialidadAdmisora",0);
         String Motivo = busca.getDato2(doc,"MotivoIngreso",0);
         request.setAttribute("Fecha",Fecha);
         request.setAttribute("Modo",Modo);
         request.setAttribute("Especialidad",Especialidad);
         request.setAttribute("Motivo",Motivo);
         // Por último almacena el documento
         resource.setContentAsDOM(doc);
         col.storeResource(resource);
         }
```

```
        catch (IOException e) {
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

EditDemograficos.java

```
/*
 * EditDemograficos.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
```

```
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase EditDemograficos, esta clase sera la encargada
 * de recuperar los datos del documento XML almacenado y mandarlos a
 * la página JSP correspondiente al formulario "Datos
 * demograficos", todo ello mediante Dom
 */
public class EditDemograficos extends Action {
   public boolean editDemo(HttpServletRequest request, HttpServletResponse
response ) throws ServletException, IOException {
       Collection col = null;
       try {
          // Consigue una instancia collection
          col = getCollection(request,response);
          // Recupera el dockey del paciente que estamos rellenando
          String dockey = request.getParameter("DOCKEY");
```

```
// Recuperar un documento de una coleccion:
XMLResource resource =(XMLResource) col.getResource(dockey);
// Convierte a objeto Dom
Document doc = (Document) resource.getContentAsDOM();
// Leo los datos del documento y los almacena en strings
// para después enviarlos a la página JSP, para ello
// usamos las funciones definidas en la clase BuscaNodos
BuscaNodos busca = new BuscaNodos();
String ValorIdhc = busca.getDato1(doc,"ValorId",1);
String ValorIdss = busca.getDato1(doc,"ValorId",2);
String FechaNacimiento = busca.getDato1(doc,"FechaNacimiento",0);
String Sexo = busca.getDato1(doc,"Sexo",0);
String Calle = busca.getDato1(doc,"Calle",0);
String Numero = busca.getDato1(doc,"Numero",0);
String Piso = busca.getDato1(doc,"Piso",0);
String Localidad = busca.getDato1(doc,"Localidad",0);
String Provincia = busca.getDato1(doc,"Provincia",0);
String TipoDireccion = busca.getDato1(doc,"TipoDireccion",0);
String CodigoPostal = busca.getDato1(doc,"CodigoPostal",0);
String tfno1 = busca.getDato1(doc,"TelecomNoEstructurado",0);
String clase1 = busca.getDato1(doc,"ClaseTelecom",0);
String tipo1 = busca.getDato1(doc,"TipoTelecom",0);
String tfno2 = busca.getDato1(doc,"TelecomNoEstructurado",1);
String clase2 = busca.getDato1(doc,"ClaseTelecom",1);
String tipo2 = busca.getDato1(doc,"TipoTelecom",1);
String tfno3 = busca.getDato1(doc,"TelecomNoEstructurado",2);
String clase3 = busca.getDato1(doc,"ClaseTelecom",2);
```

```
        String tipo3 = busca.getDato1(doc,"TipoTelecom",2);
        request.setAttribute("ValorIdhc",ValorIdhc);
        request.setAttribute("ValorIdss",ValorIdss);
        request.setAttribute("FechaNacimiento",FechaNacimiento);
        request.setAttribute("Sexo",Sexo);
        request.setAttribute("Calle",Calle);
        request.setAttribute("Numero",Numero);
        request.setAttribute("Piso",Piso);
        request.setAttribute("Localidad",Localidad);
        request.setAttribute("Provincia",Provincia);
        request.setAttribute("TipoDireccion",TipoDireccion);
        request.setAttribute("CodigoPostal",CodigoPostal);
        request.setAttribute("tfno1",tfno1);
        request.setAttribute("clase1",clase1);
        request.setAttribute("tipo1",tipo1);
        request.setAttribute("tfno2",tfno2);
        request.setAttribute("clase2",clase2);
        request.setAttribute("tipo2",tipo2);
        request.setAttribute("tfno3",tfno3);
        request.setAttribute("clase3",clase3);
        request.setAttribute("tipo3",tipo3);
        // Por último almacena el documento
        resource.setContentAsDOM(doc);
        col.storeResource(resource);
        }
      catch (IOException e) {
        System.out.println(e);
```

```
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

EditExploracion.java

```
/*
 * EditExploracion.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
```

```
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase EditExploracion, esta clase sera la encargada
 * de recuperar los datos del documento XML almacenado y mandarlos a
 * la página JSP correspondiente al formulario "Exploracion fisica",
 * todo ello mediante Dom
 */
public class EditExploracion extends Action {
  public boolean editExp(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
      Collection col = null;
      try {
          // Consigue una instancia collection
          col = getCollection(request,response);
          // Recupera el dockey del paciente que estamos rellenando
          String dockey = request.getParameter("DOCKEY");
          // Recuperar un documento de una coleccion:
          XMLResource resource =(XMLResource) col.getResource(dockey);
```

```
// Convierte a objeto Dom
Document doc = (Document) resource.getContentAsDOM();
// Leo los datos del documento y los almacena en strings para
// después enviarlos a la página JSP, para ello usamos las
// funciones definidas en la clase BuscaNodos
BuscaNodos busca = new BuscaNodos();
String Fecha = busca.getDato1(doc,"FechaTest",0);
String Peso = busca.getDato1(doc,"ValorCantidad",0);
String FC = busca.getDato1(doc,"ValorCantidad",1);
String Temp = busca.getDato1(doc,"ValorCantidad",2);
String Talla = busca.getDato1(doc,"ValorCantidad",3);
String TA = busca.getDato1(doc,"ValorCantidad",4);
String Aspecto = busca.getDato1(doc,"DescripcionMedida",0);
String Cabeza = busca.getDato1(doc,"DescripcionMedida",1);
String Torax = busca.getDato1(doc,"DescripcionMedida",2);
String Abdomen = busca.getDato1(doc,"DescripcionMedida",3);
String Extremidades = busca.getDato1(doc,"DescripcionMedida",4);
// En el caso de la etiqueta comentario, tenemos que tener
// en cuenta, que depende del número de procedimientos que se
// hayan introducido, por lo que tenemos que hayar dicho numero
NodeList listaproc = doc.getElementsByTagName("Procedimiento");
int numproc = listaproc.getLength();
String Observ = busca.getDato1(doc,"Comentario",numproc);
request.setAttribute("Fecha",Fecha);
request.setAttribute("Peso",Peso);
request.setAttribute("FC",FC);
request.setAttribute("Temp",Temp);
```

```
            request.setAttribute("Talla",Talla);
            request.setAttribute("TA",TA);
            request.setAttribute("Aspecto",Aspecto);
            request.setAttribute("Cabeza",Cabeza);
            request.setAttribute("Torax",Torax);
            request.setAttribute("Abdomen",Abdomen);
            request.setAttribute("Extremidades",Extremidades);
            request.setAttribute("Observ",Observ);
            // Almacenar el documento
            resource.setContentAsDOM(doc);
            col.storeResource(resource);
            }
        catch (IOException e) {
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
            if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
```

```
        }
        return true;
    }
}
```

EditHemograma.java

```
/*
 * EditHemograma.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase EditHemograma, esta clase sera la encargada
 * de recuperar los datos del documento XML almacenado y mandarlos a
```

```
 * la página JSP correspondiente al formulario "Hemograma",
 * todo ello mediante Dom
 */
public class EditHemograma extends Action {
  public boolean editHem(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
      Collection col= null;
      try{
         // Consigue una instancia collection
         col = getCollection(request,response);
         // Recupera el dockey del paciente que estamos rellenando
         String dockey = request.getParameter("DOCKEY");
         // Recuperar un documento de una coleccion:
         XMLResource resource =(XMLResource) col.getResource(dockey);
         // Convierte a objeto Dom
         Document doc = (Document) resource.getContentAsDOM();
         // Leo los datos del documento y los almacena en strings para
         // después enviarlos a la página JSP, para ello usamos las
         // funciones definidas en la clase BuscaNodos
         BuscaNodos busca = new BuscaNodos();
         String Fecha = busca.getDato1(doc,"FechaTest",1);
         String Hematies = busca.getDato1(doc,"ValorCantidad",5);
         String Hemoglobina = busca.getDato1(doc,"ValorCantidad",6);
         String Hematocrito = busca.getDato1(doc,"ValorCantidad",7);
         String VCM = busca.getDato1(doc,"ValorCantidad",8);
         String HCM = busca.getDato1(doc,"ValorCantidad",9);
         String CHCM = busca.getDato1(doc,"ValorCantidad",10);
         String RDW = busca.getDato1(doc,"ValorCantidad",11);
```

```
String Leucocitos = busca.getDato1(doc,"ValorCantidad",12);
String Segmentados = busca.getDato1(doc,"ValorCantidad",13);
String Linfocitos = busca.getDato1(doc,"ValorCantidad",14);
String Monocitos = busca.getDato1(doc,"ValorCantidad",15);
String Eosinofilos = busca.getDato1(doc,"ValorCantidad",16);
String Basofilos = busca.getDato1(doc,"ValorCantidad",17);
String Plaquetas = busca.getDato1(doc,"ValorCantidad",18);
// En el caso de la etiqueta comentario, tenemos que tener
// en cuenta, que depende del número de procedimientos que se
// hayan introducido, por lo que tenemos que hayar dicho numero
NodeList listaproc = doc.getElementsByTagName("Procedimiento");
int numproc = listaproc.getLength();
String Observ = busca.getDato1(doc,"Comentario",numproc+1);
request.setAttribute("Fecha",Fecha);
request.setAttribute("Hematies",Hematies);
request.setAttribute("Hemoglobina",Hemoglobina);
request.setAttribute("Hematocrito",Hematocrito);
request.setAttribute("VCM",VCM);
request.setAttribute("HCM",HCM);
request.setAttribute("CHCM",CHCM);
request.setAttribute("RDW",RDW);
request.setAttribute("Leucocitos",Leucocitos);
request.setAttribute("Segmentados",Segmentados);
request.setAttribute("Linfocitos",Linfocitos);
request.setAttribute("Monocitos",Monocitos);
request.setAttribute("Eosinofilos",Eosinofilos);
request.setAttribute("Basofilos",Basofilos);
```

```
            request.setAttribute("Plaquetas",Plaquetas);
            request.setAttribute("ObservHemo",Observ);
            // Almacenar el documento
            resource.setContentAsDOM(doc);
            col.storeResource(resource);
        }
        catch (IOException e) {
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
           if (e.getMessage() != null ) {
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

EditInforme.java

```
/*
 * EditInforme.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import java.net.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase EditInforme, esta clase es muy parecida a
 * AddInforme, la única diferencia reside en que en este servlet el
 * documento referente al paciente ya existe, y no hay que crearlo,
 * asi que recuperaremos los datos del formulario y los almacenaremos
 * en su documento correspondiente
 */
public class EditInforme extends Action {
```

```
  public boolean edit(HttpServletRequest request, HttpServletResponse response )  throws ServletException, IOException {

      Collection col = null;

      String ourdoc = null;

      try {

        // Consigue la instancia collection

        col = getCollection(request,response);

        // Recupera los parametros del formulario almacenandolos como strings

        String dockey = request.getParameter("DOCKEY");

        String DNI = request.getParameter("VALORID");

        String Nombre = request.getParameter("NOMBRE");

        String PrimerApellido = request.getParameter("PRIMERAPELLIDO");

        String SegundoApellido = request.getParameter("SEGUNDOAPELLIDO");

        // Crea el string XML con los datos del formulario

        ourdoc = toXml(DNI,Nombre,PrimerApellido,SegundoApellido);

        // Toma XMLResource y sustituye su contenido

        XMLResource resource = (XMLResource) col.getResource(dockey);

        resource.setContent(ourdoc);

        col.storeResource(resource);

      } catch ( Exception e) {

        e.printStackTrace();

        // Si la respuesta es committed no hay mucho más que se pueda hacer

        if (response.isCommitted())

            return true;

        // Toma la excepción y envia al usuario a la página de error

        if (e.getMessage() != null ) {

          response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
```

```
        }
        else {
           response.sendRedirect("/InformeAlta/error.jsp" );
        }
      }
      return true;
   }
}
```

EditTratamiento.java

```
/*
 * EditTratamiento.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
```

```
 */
/**
 * Definimos la clase EditTratamiento, esta clase sera la encargada
 * de recuperar los datos del documento XML almacenado y mandarlos a
 * la página JSP correspondiente al formulario "Tratamiento y Alta",
 * todo ello mediante Dom
 */
public class EditTratamiento extends Action {
    public boolean editTra(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
        Collection col= null;
        try{
            // Consigue una instancia collection
            col = getCollection(request,response);
            // Recupera el dockey del paciente que estamos rellenando
            String dockey = request.getParameter("DOCKEY");
            // Recuperar un documento de una coleccion:
            XMLResource resource =(XMLResource) col.getResource(dockey);
            // Convierte a objeto Dom
            Document doc = (Document) resource.getContentAsDOM();
            // Leo los datos del documento y los almacena en strings para
            // después enviarlos a la página JSP, para ello usamos las
            // funciones definidas en la clase BuscaNodos
            BuscaNodos busca = new BuscaNodos();
            String Fecha = busca.getDato1(doc,"FechaAlta",0);
            String Tipo = busca.getDato1(doc,"TipoAlta",0);
            String Lugar = busca.getDato1(doc,"LugarAlta",0);
            String Trat = busca.getDato1(doc,"DescripcionMedicamento",0);
```

```
            String Adm = busca.getDato1(doc,"DosisNoEstructurada",0);
            // En el caso de la etiqueta comentario, tenemos que tener
            // en cuenta, que depende del número de procedimientos que se
            // hayan introducido, por lo que tenemos que hayar dicho numero
            NodeList listaproc = doc.getElementsByTagName("Procedimiento");
            int numproc = listaproc.getLength();
            String ObservTrat = busca.getDato1(doc,"Comentario",numproc+3);
            request.setAttribute("Fecha",Fecha);
            request.setAttribute("Tipo",Tipo);
            request.setAttribute("Lugar",Lugar);
            request.setAttribute("Trat",Trat);
            request.setAttribute("Adm",Adm);
            request.setAttribute("ObservTrat",ObservTrat);
            // Almacenar el documento
            resource.setContentAsDOM(doc);
            col.storeResource(resource);
        }
        catch (IOException e) {
            System.out.println(e);
        }
        catch ( Exception e) {
            e.printStackTrace();
            // Si la respuesta es committed no hay mucho más que se pueda hacer
            if (response.isCommitted())
                return true;
            // Toma la excepción y envia al usuario a la página de error
           if (e.getMessage() != null ) {
```

```
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

Group.java

```
/*
 * Group.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.util.Vector;
import java.util.Hashtable;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.w3c.dom.*;
import org.apache.xpath.*;
import org.apache.xpath.objects.*;
/**
```

```
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase Group, que nos ayudará a trabajar con los grupos de
 * clases del tipo Informe
 */
public class Group {
   private Vector informeList = new Vector();
   private int index = 0;
   /**
    * Devuelve el index actual
    */
   public int getIndex() {
      return index - 1;
   }
   /**
    * Devuelve el siguiente informe de la lista
    */
   public Informe getNext() {
      Informe informe = (Informe) informeList.get(index);
      index = index + 1;
      return informe;
   }
   /**
    * Devuelve el elemento informe con un index determinado
    */
   public Informe getInformeAt(int i) {
```

```
    return (Informe) informeList.get(i);
  }
  /**
   * Añade informe al Group
   */
  public void add(Informe informe) {
    informeList.add(informe);
  }
  /**
   * Añade Informes a partir de Result set
   */
  public void addResults(ResourceIterator results) {
    try {
      while (results.hasMoreResources()) {
          // Crea una nueva instancia informe para añadir al Group
          Informe informe = new Informe();
          // Recupera el nodo siguiente
          XMLResource resource = (XMLResource) results.nextResource();
          // Lo convierte a nodo Dom
          Node originalnode = resource.getContentAsDOM();
          // Consigue el dockey de este nodo para su posterior uso
          String dockey = resource.getDocumentId();
          informe.setDocKey(dockey);
          // Usa parser xpath Xalan para extraer los datos, almacenados
          // en una hastable
          XObject xo = new XObject();
          xo = XPathAPI.eval(originalnode,
"/informe/DatosDemograficosPaciente/IdPaciente");
```

```
            // DNI
            xo = XPathAPI.eval(originalnode,
"/informe/DatosDemograficosPaciente/IdPaciente/ValorId");
            informe.setValorId(xo.toString());
            // Nombre y apellidos del paciente
            xo = XPathAPI.eval(originalnode,
"/informe/DatosDemograficosPaciente/NombrePaciente/NombreEstructurado/N
ombre");
            informe.setNombre(xo.toString());
            xo = XPathAPI.eval(originalnode,
"/informe/DatosDemograficosPaciente/NombrePaciente/NombreEstructurado/Pr
imerApellido");
            informe.setPrimerApellido(xo.toString());
            xo = XPathAPI.eval(originalnode,
"/informe/DatosDemograficosPaciente/NombrePaciente/NombreEstructurado/S
egundoApellido");
            informe.setSegundoApellido(xo.toString());
            // Fecha de Alta
            xo = XPathAPI.eval(originalnode,
"/informe/AdministracionAltaPaciente/DatosAlta/FechaAlta");
            informe.setFechaAlta(xo.toString());
            // Añade un informe al group
            this.add(informe);
        }
    }
    catch (Exception e) {
        e.printStackTrace();
    }
  }
  /**
   * Borra un informe del Group
```

```
   */
  public void remove(int i) {
     informeList.remove(i);
  }
  /**
   * Borra todos los informess del Group
   */
  public void removeAll() {
     informeList.removeAllElements();
  }
  /**
   * Return the size of the Group
   */
  public int sizeOf() {
     return informeList.size();
  }
  /**
   * Borra el contador para recuperar informe
   */
  public void reset() {
     index = 0;
  }
  /**
   * Devuelve true si existen mas informes
   */
  public boolean hasMoreInformes() {
     return index < informeList.size();
```

```
  }
}
```

Informe.java

```
/*
 * Informe.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase Informe, que nos permitirá mantener los datos de un
 * contacto
 */
public class Informe {
    String dockey = null;
    String ValorId = null;
    String Nombre = null;
    String PrimerApellido = null;
    String SegundoApellido = null;
    String FechaAlta = null;
    // Document Key
    public void setDocKey(String key) {
```

```
        dockey = key;
    }
    public String getDocKey() {
        return dockey;
    }
    // ValorId: DNI
    public void setValorId(String valor){
        ValorId = valor;
    }
    public String getValorId(){
        return ValorId;
    }
     // Nombre
    public void setNombre(String name) {
        Nombre = name;
    }
    public String getNombre() {
        return Nombre;
    }
    // Primer Apellido
    public void setPrimerApellido(String name) {
        PrimerApellido = name;
    }
    public String getPrimerApellido() {
        return PrimerApellido;
    }
    // Segundo Apellido
```

```
    public void setSegundoApellido(String name) {
        SegundoApellido = name;
    }
    public String getSegundoApellido() {
        return SegundoApellido;
    }
    // Fecha Alta
    public void setFechaAlta(String name) {
        FechaAlta = name;
    }
    public String getFechaAlta() {
        return FechaAlta;
    }
}
```

ListInformes.java

```
/*
 * ListInformes.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
```

```
import org.xmldb.api.modules.*;
import java.net.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase ListInformes, esta clase sera la encargada
 * de mostrar una lista con todos los informes almacenados en la base
 * de datos, o de mostrar un mensaje en caso de que la base de datos esté
 * vacia
 */
public class ListInformes extends Action {
  public boolean list(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
     Collection col = null;
     try {
       // Consigue una referencia de la sesión
       HttpSession session = request.getSession(true);
       // Consigue una instancia collecion
       col = getCollection(request,response);
       // Recurea la instancia de Group para esta sesión
       Group group = (Group)session.getAttribute("group");
       XPathQueryService service = (XPathQueryService)col.getService("XPathQueryService",XMLDBAPIVERSION);
       // Consigue todos los elementos informe de la base de datos
       ResourceSet resultSet = service.query("/informe");
       ResourceIterator results = resultSet.getIterator();
       // Limpia el objeto group...
```

```
        group.removeAll();
        // Añade los resultados de la búsqueda Xpath a la instancia Group
        group.addResults(results);
      } catch (Exception e) {
        e.printStackTrace();
         // Si la respuesta es committed no hay mucho más que se pueda hacer
          if (response.isCommitted())
              return true;
          // Toma la excepción y envia al usuario a la página de error
         if (e.getMessage() != null ) {
              response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
          }
          else {
              response.sendRedirect("/InformeAlta/error.jsp" );
          }
       }
       return true;
   }
}
```

ListJuicios.java

```
/*
 * ListJuicios.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
```

```
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase ListJuicios, esta clase sera la encargada
 * de recuperar los datos del documento XML referentes a los
 * juicios y mandarlos a la página JSP correspondiente al
 * formulario "Juicios Clínicos", todo ello mediante Dom
 */
public class ListJuicios extends Action {
  public boolean listJui(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
     Collection col = null;
     try {
          // Consigue una instancia collection
          col = getCollection(request,response);
          // Recupera el dockey del paciente que estamos rellenando
          String dockey = request.getParameter("DOCKEY");
```

```
        // Recuperar un documento de una coleccion:
        XMLResource resource =(XMLResource) col.getResource(dockey);
        // Convierte a objeto Dom
        Document doc = (Document) resource.getContentAsDOM();
        // Calculo el numero de Juicios que existe
        NodeList listadiag = doc.getElementsByTagName("Patologia");
        int numdiag = listadiag.getLength();
        // Tambien necesito saber el numero de Procedimientos
        // almacenados
        NodeList listaproc = doc.getElementsByTagName("Procedimiento");
        int numproc = listaproc.getLength();
        // Leo los datos del documento y los almacena en strings para
        // después enviarlos a la página JSP, para ello usamos las
        // funciones definidas en la clase BuscaNodos
        BuscaNodos busca = new BuscaNodos();
        for(int i=0;i<numdiag;i++){
            int x = 3+numproc;
            if(i==0){
                String Id0 =
busca.getDato1(doc,"IdEsquemaCodigoClinico",i+numproc);
                String Nombre0 =
busca.getDato1(doc,"DescripcionCodigoClinico",i+numproc);
                String Caracter0 =
busca.getDato1(doc,"ImportanciaDiagnostico",i);
                request.setAttribute("Id0",Id0);
                request.setAttribute("Nombre0",Nombre0);
                request.setAttribute("Caracter0",Caracter0);
            }
            if(i==1){
```

```
            String                                  Id1                                 =
busca.getDato1(doc,"IdEsquemaCodigoClinico",i+numproc);

            String                                  Nombre1                             =
busca.getDato1(doc,"DescripcionCodigoClinico",i+numproc);

            String                                  Caracter1                           =
busca.getDato1(doc,"ImportanciaDiagnostico",i);

            request.setAttribute("Id1",Id1);

            request.setAttribute("Nombre1",Nombre1);

            request.setAttribute("Caracter1",Caracter1);

        }

        if(i==2){

            String                                  Id2                                 =
busca.getDato1(doc,"IdEsquemaCodigoClinico",i+numproc);

            String                                  Nombre2                             =
busca.getDato1(doc,"DescripcionCodigoClinico",i+numproc);

            String                                  Caracter2                           =
busca.getDato1(doc,"ImportanciaDiagnostico",i);

            request.setAttribute("Id2",Id2);

            request.setAttribute("Nombre2",Nombre2);

            request.setAttribute("Caracter2",Caracter2);

        }

        if(i==3){

            String                                  Id3                                 =
busca.getDato1(doc,"IdEsquemaCodigoClinico",i+numproc);

            String                                  Nombre3                             =
busca.getDato1(doc,"DescripcionCodigoClinico",i+numproc);

            String                                  Caracter3                           =
busca.getDato1(doc,"ImportanciaDiagnostico",i);

            request.setAttribute("Id3",Id3);

            request.setAttribute("Nombre3",Nombre3);

            request.setAttribute("Caracter3",Caracter3);
```

```
            }
          }
          // Almacenar el documento
          resource.setContentAsDOM(doc);
          col.storeResource(resource);
        }
        catch (IOException e) {
          System.out.println(e);
        }
        catch ( Exception e) {
          e.printStackTrace();
          // Si la respuesta es committed no hay mucho más que se pueda hacer
          if (response.isCommitted())
            return true;
          // Toma la excepción y envia al usuario a la página de error
          if (e.getMessage() != null ) {
            response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
          }
          else {
            response.sendRedirect("/InformeAlta/error.jsp" );
          }
        }
        return true;
    }
}
```

ListPatologias.java

```
/*
 * ListPatologias.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase ListPatologia, esta clase sera la encargada
 * de recuperar los datos del documento XML referentes a las
 * patologias y mandarlos a la página JSP correspondiente al
 * formulario "Patologias previas", todo ello mediante Dom
 */
public class ListPatologias extends Action {
```

```
  public boolean listPat(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
     Collection col = null;
     try {
         // Consigue una instancia collection
          col = getCollection(request,response);
          // Recupera el dockey del paciente que estamos rellenando
          String dockey = request.getParameter("DOCKEY");
          // Recuperar un documento de una coleccion:
          XMLResource resource =(XMLResource) col.getResource(dockey);
          // Convierte a objeto Dom
          Document doc = (Document) resource.getContentAsDOM();
          // Calculo el numero de patologias que existe
          NodeList listapat = doc.getElementsByTagName("RegistroPatologia");
          int numpat = listapat.getLength();
          // Tambien necesito saber el numero de Juicios y
          // Procedimientos almacenados
          NodeList listadiag = doc.getElementsByTagName("Patologia");
          int numdiag = listadiag.getLength();
          NodeList listaproc = doc.getElementsByTagName("Procedimiento");
          int numproc = listaproc.getLength();
          // Leo los datos del documento y los almacena en strings para
          // después enviarlos a la página JSP, para ello usamos las
          // funciones definidas en la clase BuscaNodos
          BuscaNodos busca = new BuscaNodos();
          for(int i=0;i<numpat;i++){
              if(i==0){
```

```
            String Id0 = busca.getDato1(doc,"IdEsquemaCodigoClinico",i+numdiag+numproc);
            String Nombre0 = busca.getDato1(doc,"DescripcionCodigoClinico",i+numdiag+numproc);
            String Fecha0 = busca.getDato1(doc,"FechaDiagnostico",i);
            request.setAttribute("Id0",Id0);
            request.setAttribute("Nombre0",Nombre0);
            request.setAttribute("Fecha0",Fecha0);
        }
        if(i==1){
            String Id1 = busca.getDato1(doc,"IdEsquemaCodigoClinico",i+numdiag+numproc);
            String Nombre1 = busca.getDato1(doc,"DescripcionCodigoClinico",i+numdiag+numproc);
            String Fecha1 = busca.getDato1(doc,"FechaDiagnostico",i);
            request.setAttribute("Id1",Id1);
            request.setAttribute("Nombre1",Nombre1);
            request.setAttribute("Fecha1",Fecha1);
        }
        if(i==2){
            String Id2 = busca.getDato1(doc,"IdEsquemaCodigoClinico",i+numdiag+numproc);
            String Nombre2 = busca.getDato1(doc,"DescripcionCodigoClinico",i+numdiag+numproc);
            String Fecha2 = busca.getDato1(doc,"FechaDiagnostico",i);
            request.setAttribute("Id2",Id2);
            request.setAttribute("Nombre2",Nombre2);
            request.setAttribute("Fecha2",Fecha2);
        }
        if(i==3){
```

```
                String Id3 = busca.getDato1(doc,"IdEsquemaCodigoClinico",i+numdiag+numproc);
                String Nombre3 = busca.getDato1(doc,"DescripcionCodigoClinico",i+numdiag+numproc);
                String Fecha3 = busca.getDato1(doc,"FechaDiagnostico",i);
                request.setAttribute("Id3",Id3);
                request.setAttribute("Nombre3",Nombre3);
                request.setAttribute("Fecha3",Fecha3);
            }
        }
        // Almacenar el documento
        resource.setContentAsDOM(doc);
        col.storeResource(resource);
    }
    catch (IOException e) {
        System.out.println(e);
    }
    catch ( Exception e) {
        e.printStackTrace();
        // Si la respuesta es committed no hay mucho más que se pueda hacer
        if (response.isCommitted())
            return true;
        // Toma la excepción y envia al usuario a la página de error
        if (e.getMessage() != null ) {
            response.sendRedirect("/InformeAlta/error.jsp?error=" + URLEncoder.encode(e.getMessage()) );
        }
        else {
            response.sendRedirect("/InformeAlta/error.jsp" );
```

```
            }
        }
        return true;
    }
}
```

ListProcedimientos.java

```
/*
 * Listprocedimientos.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase ListProcedimientos, esta clase sera la encargada
```

```
 * de recuperar los datos del documento XML referentes a los
 * procedimientos y mandarlos a la página JSP correspondiente al
 * formulario "Procedimientos realizados", todo ello mediante Dom
 */
public class ListProcedimiento extends Action {
  public boolean listProc(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
     Collection col = null;
     try {
        // Consigue una instancia collection
         col = getCollection(request,response);
         // Recupera el dockey del paciente que estamos rellenando
         String dockey = request.getParameter("DOCKEY");
         // Recuperar un documento de una coleccion:
         XMLResource resource =(XMLResource) col.getResource(dockey);
         // Convierte a objeto Dom
         Document doc = (Document) resource.getContentAsDOM();
         // Calculo el numero de procedimientos que existe
         NodeList listaproc = doc.getElementsByTagName("Procedimiento");
         int numproc = listaproc.getLength();
         // Leo los datos del documento y los almacena en strings para
         // después enviarlos a la página JSP, para ello usamos las
         // funciones definidas en la clase BuscaNodos
         BuscaNodos busca = new BuscaNodos();
         for(int i=0;i<numproc;i++){
            if(i==0){
               String Id0 = busca.getDato1(doc,"IdEsquemaCodigoClinico",i);
```

```
            String Nombre0 = busca.getDato1(doc,"DescripcionCodigoClinico",i);
            String Comentario0 = busca.getDato1(doc,"Comentario",i);
            request.setAttribute("Id0",Id0);
            request.setAttribute("Nombre0",Nombre0);
            request.setAttribute("Comentario0",Comentario0);
        }
        if(i==1){
            String Id1 = busca.getDato1(doc,"IdEsquemaCodigoClinico",i);
            String Nombre1 = busca.getDato1(doc,"DescripcionCodigoClinico",i);
            String Comentario1 = busca.getDato1(doc,"Comentario",i);
            request.setAttribute("Id1",Id1);
            request.setAttribute("Nombre1",Nombre1);
            request.setAttribute("Comentario1",Comentario1);
        }
        if(i==2){
            String Id2 = busca.getDato1(doc,"IdEsquemaCodigoClinico",i);
            String Nombre2 = busca.getDato1(doc,"DescripcionCodigoClinico",i);
            String Comentario2 = busca.getDato1(doc,"Comentario",i);
            request.setAttribute("Id2",Id2);
            request.setAttribute("Nombre2",Nombre2);
            request.setAttribute("Comentario2",Comentario2);
        }
        if(i==3){
            String Id3 = busca.getDato1(doc,"IdEsquemaCodigoClinico",i);
            String Nombre3 = busca.getDato1(doc,"DescripcionCodigoClinico",i);
```

```
                String Comentario3 = busca.getDato1(doc,"Comentario",i);
                request.setAttribute("Id3",Id3);
                request.setAttribute("Nombre3",Nombre3);
                request.setAttribute("Comentario3",Comentario3);
            }
        }
        // Almacenar el documento
        resource.setContentAsDOM(doc);
        col.storeResource(resource);
    }
    catch (IOException e) {
        System.out.println(e);
    }
    catch ( Exception e) {
        e.printStackTrace();
        // Si la respuesta es committed no hay mucho más que se pueda hacer
        if (response.isCommitted())
            return true;
        // Toma la excepción y envia al usuario a la página de error
       if (e.getMessage() != null ) {
            response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
        }
        else {
            response.sendRedirect("/InformeAlta/error.jsp" );
        }
    }
    return true;
```

```
    }
}
```

SearchInforme.java

```
/*
 * SearchInforme.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.util.Vector;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import java.net.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase SearchInforme para hacer búsquedas en la base de
 * datos usando Xpath
 */
public class SearchInforme extends Action {
```

```
  public boolean attributeSearch(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {

      Collection col = null;

      Vector vector = null;

      String xpath = null;

      try {

        HttpSession session = request.getSession(true);

        // Get a collection instance

        col = getCollection(request,response);

        XPathQueryService service = (XPathQueryService)col.getService("XPathQueryService",XMLDBAPIVERSION);

        // Get the seach parameters from the form

        String searchtype = request.getParameter("SEARCHTYPE");

        String searchstring = request.getParameter("SEARCHSTRING");

        // Setup xpath string depending on search type

        if ( searchtype.equals("ValorId") ) {

          xpath = "/informe/DatosDemograficosPaciente/IdPaciente[" + searchtype + "='" + searchstring + "']";

        } else if ( searchtype.equals("Nombre") || searchtype.equals("PrimerApellido") || searchtype.equals("SegundoApellido")) {

          xpath = "/informe/DatosDemograficosPaciente/NombrePaciente/NombreEstructurado[" + searchtype + "='" + searchstring + "']";

        } else if ( searchtype.equals("FechaAdmision")) {

          xpath = "/informe/AdministracionAltaPaciente/DatosAdmision[" + searchtype + "='" + searchstring + "']";

        }

        ResourceSet resultSet = service.query(xpath);

        ResourceIterator results = resultSet.getIterator();

        Group group = (Group)session.getAttribute("group");
```

```
        // Limpia el objeto group...
        group.removeAll();
        // Añade los resultados de la búsqueda Xpath a la instancia Group
        group.addResults(results);
      } catch(Exception e) {
        e.printStackTrace();
         // there's not much else we can do if the response is committed
         if (response.isCommitted())
            return true;
        // Catch the exception and send the user to the error page
        if (e.getMessage() != null ) {
          response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
        }
        else {
          response.sendRedirect("/InformeAlta/error.jsp" );
        }
      }
      return true;
   }
   public boolean xpathSearch(HttpServletRequest request,
HttpServletResponse response ) throws ServletException, IOException {
      Collection col = null;
      Vector vector = null;
      String xpath = null;
      try {
        HttpSession session = request.getSession(true);
        // Get a collection instance
```

```
        col = getCollection(request,response);
        XPathQueryService service =
(XPathQueryService)col.getService("XPathQueryService",XMLDBAPIVERSION
);
        // Get the seach parameters from the form
        String searchstring = request.getParameter("SEARCHSTRING");
        ResourceSet resultSet = service.query(searchstring);
        ResourceIterator results = resultSet.getIterator();
        Group group = (Group)session.getAttribute("group");
        // Clear out group object...
        group.removeAll();
        // Add results to Group instance
        group.addResults(results);
      } catch(Exception e) {
        e.printStackTrace();
         // there's not much else we can do if the response is committed
         if (response.isCommitted())
            return true;
        // Catch the exception and send the user to the error page
        if (e.getMessage() != null ) {
          response.sendRedirect("/InformeAlta/error.jsp?error="
URLEncoder.encode(e.getMessage()) );
        }
        else {
          response.sendRedirect("/InformeAlta/error.jsp" );
        }
      }
      return true;
    }
```

```
}
```

ShowXML.java

```
/*
 * ShowXML.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.DatabaseManager;
import org.w3c.dom.*;
/**
 * @author  Irene Lavado Gomez
 */
/**
 * Definimos la clase ShowXML, que muestra el archivo XML correspondiente
 * al informe de alta de un paciente
 */
public class ShowXML extends Action {
```

```
public boolean showXml(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
    Collection col= null;
    try{
        // Consigue una instancia collecion
        col = getCollection(request,response);
        // Toma el dockey del paciente que estamos rellenando
        String dockey = request.getParameter("DOCKEY");
        // Recuperar un documento de una coleccion
        XMLResource resource =(XMLResource) col.getResource(dockey);
        // Almaceno en file el documento xml correspondiente al dockey
        String file = (String) resource.getContent();
        response.setContentType("text/xml");
        java.io.PrintWriter out = response.getWriter();
        out.println(file);
        out.close();
    }
    catch (IOException e) {
        System.out.println(e);
    }
    catch ( Exception e) {
        e.printStackTrace();
        // Si la respuesta es committed no hay mucho más que se pueda hacer
        if (response.isCommitted())
            return true;
        // Toma la excepción y envia al usuario a la página de error
        if (e.getMessage() != null ) {
```

```
                response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
            }
            else {
                response.sendRedirect("/InformeAlta/error.jsp" );
            }
        }
        return true;
    }
}
```

Task.java

```
/*
 * Task.java
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 */
package InformeAlta;
import java.io.*;
import java.net.*;
import java.beans.*;
import javax.servlet.*;
import javax.servlet.jsp.*;
import javax.servlet.http.*;
/**
 * @author  Irene Lavado Gomez
```

```
 */

/**

 * Definimos la clase Task, esta clase actua como un controlador,

 * enviando la respuesta HTML a la correcta clase o pagina JSP

 * dependiendo del valor de action

 */

public class Task extends HttpServlet {

  public void doPost(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {

     JspFactory jspFactory = null;

     PageContext pageContext = null;

     HttpSession session = null;

     InformeAlta.Group group = null;

     String ind = request.getParameter("INDEX");

     String action = null;

     try {

       // Si el bean Group no ha sido añadido aun a la seseion, lo

       // añadimos, esto permitira al usuario ir directamente a

       // cualquier pagina

       jspFactory = JspFactory.getDefaultFactory();

       pageContext = jspFactory.getPageContext(this, request, response, "/error.jsp", true, 8192, true );

       session = pageContext.getSession();

       group = (Group)pageContext.getAttribute("group",PageContext.SESSION_SCOPE);

       if (group == null) {


group=(InformeAlta.Group)Beans.instantiate(this.getClass().getClassLoader(), "InformeAlta.Group" );
```

```
        pageContext.setAttribute("group",                    group,
PageContext.SESSION_SCOPE );
      }
      // Comprobamos que action es distinto de null
      if ( request.getParameter("action") != null ) {
        action = request.getParameter("action");
      }
      else {
        action = request.getParameter("ACTION");
      }
      // Decidimos donde enviar al usuario dependiendo del valor
      // de action
      if ( action != null ) {
        // Añadir un paciente
        if ( action.equalsIgnoreCase("addinf") ) {
          gotoPage("/addInforme.jsp",request,response);
        } else if ( action.equalsIgnoreCase("addinforme") ) {
          AddInforme informe = new AddInforme();
          informe.add(request,response);
          ListInformes list = new ListInformes();
          list.list(request,response);
          gotoPage("/listInformes.jsp",request,response);
         // Editar datos personales
        }   else if ( action.equalsIgnoreCase("editpersonales") ) {
           gotoPage("/editPersonales.jsp",request,response);
         // Modificar datos personales
        } else if ( action.equalsIgnoreCase("editinforme") ) {
           EditInforme informe = new EditInforme();
```

```
            informe.edit(request,response);
            ListInformes list = new ListInformes();
            list.list(request,response);
           gotoPage("/listInformes.jsp",request,response);
           // Editar los datos demograficos
          } else if ( action.equalsIgnoreCase("editdemo") ) {
            EditDemograficos dem = new EditDemograficos();
            dem.editDemo(request,response);
           gotoPage("/addDemograficos.jsp",request,response);
           // Almacenar los datos demograficos y pasar al
           // siguiente formulario
         } else if ( action.equalsIgnoreCase("adddemo") ) {
           AddDemograficos dem = new AddDemograficos();
           dem.addDemo(request,response);
           gotoPage("/addAntecedentes.jsp?index="+ind,request,response);
         // Editar antecedentes personales
        }   else if ( action.equalsIgnoreCase("addantece") ) {
            EditAntecedentes ant = new EditAntecedentes();
            ant.editAnt(request,response);
           gotoPage("/addAntecedentes.jsp",request,response);
         // Almacenar los antecedentes personales y pasas al
         // siguiente formulario
        } else if ( action.equalsIgnoreCase("addantecedentes") ) {
           AddAntecedentes ante = new AddAntecedentes();
           ante.addAnt(request,response);
           gotoPage("/addCircunstancias.jsp?index="+ind,request,response);
         // Listado de patologias
```

```
}   else if ( action.equalsIgnoreCase("addpat") ) {
     ListPatologias pat = new ListPatologias();
     pat.listPat(request,response);
    gotoPage("/addPatologias.jsp",request,response);
 // Añadir nueva patologia y volver al listado de
 // patologias
} else if ( action.equalsIgnoreCase("addpatologias") ) {
    AddPatologia pat = new AddPatologia();
    pat.addPat(request,response);
    ListPatologias patolog = new ListPatologias();
    patolog.listPat(request,response);
    gotoPage("/addPatologias.jsp?index="+ind,request,response);
// Editar circunstancias al ingreso
} else if ( action.equalsIgnoreCase("addcircunsdom") ) {
    EditCircunstancias circu = new EditCircunstancias();
    circu.editCircu(request,response);
    gotoPage("/addCircunstancias.jsp",request,response);
 // Almacenar circunstancias al ingreso y pasar al
 // siguiente formulario
} else if ( action.equalsIgnoreCase("addcircunstanciasdom") ) {
    AddCircunstanciasDom circu = new AddCircunstanciasDom();
    circu.addCircuDom(request,response);
    gotoPage("/examenes.jsp?index="+ind,request,response);
 // Examenes
} else if ( action.equalsIgnoreCase("exam") ) {
    gotoPage("/examenes.jsp",request,response);
// Editar exploracion fisica
```

```
}   else if ( action.equalsIgnoreCase("addexplor") ) {
     EditExploracion explo = new EditExploracion();
     explo.editExp(request,response);
    gotoPage("/addExploracion.jsp",request,response);
  // Almacenar datos de exploración fisica y volver a
  // examenes
} else if ( action.equalsIgnoreCase("addexploracion") ) {
    AddExploracion circu = new AddExploracion();
    circu.addExp(request,response);
    gotoPage("/examenes.jsp?index="+ind,request,response);
  // Editar hemograma
} else if ( action.equalsIgnoreCase("addhemo") ) {
     EditHemograma hemo = new EditHemograma();
     hemo.editHem(request,response);
    gotoPage("/addHemograma.jsp",request,response);
   // Almacenar datos de hemograma y volver a examenes
} else if ( action.equalsIgnoreCase("addhemograma") ) {
    AddHemograma hemo = new AddHemograma();
    hemo.addHem(request,response);
    gotoPage("/examenes.jsp?index="+ind,request,response);
  // Editar resultados de prueba bioquimica
} else if ( action.equalsIgnoreCase("addbioqui") ) {
     EditBioquimica bio = new EditBioquimica();
     bio.editBio(request,response);
    gotoPage("/addBioquimica.jsp",request,response);
   // Almacenar prueba bioquimica y volver a examenes
} else if ( action.equalsIgnoreCase("addbioquimica") ) {
```

```
            AddBioquimica circu = new AddBioquimica();
            circu.addBio(request,response);
            gotoPage("/examenes.jsp?index="+ind,request,response);
           // Listado de juicios clinicos
          }   else if ( action.equalsIgnoreCase("addjuic") ) {
             ListJuicios jui = new ListJuicios();
             jui.listJui(request,response);
             gotoPage("/addJuicio.jsp",request,response);
         // Añadir nuevo juicio y volver al listado de
         // juicios
        } else if ( action.equalsIgnoreCase("addjuicio") ) {
           AddJuicio juicio = new AddJuicio();
           juicio.addJui(request,response);
           ListJuicios jui = new ListJuicios();
            jui.listJui(request,response);
           gotoPage("/addJuicio.jsp?index="+ind,request,response);
         // Añadir comentario del juicio clinico y pasar al
         // siguiente formulario
        } else if ( action.equalsIgnoreCase("addjuiciocom") ) {
           AddJuicioComentarios juicom = new AddJuicioComentarios();
           juicom.addJuiCom(request,response);
           gotoPage("/addProcedimiento.jsp?index="+ind,request,response);
         // Listar procedimientos realizados
          }   else if ( action.equalsIgnoreCase("addproce") ) {
             ListProcedimiento pro = new ListProcedimiento();
             pro.listProc(request,response);
           gotoPage("/addProcedimiento.jsp",request,response);
```

```
// Añadir nuevo procedimiento y volver al listado de
// procedimientos
} else if ( action.equalsIgnoreCase("addprocedimiento") ) {
  AddProcedimiento proce = new AddProcedimiento();
  proce.addProc(request,response);
  ListProcedimiento pro = new ListProcedimiento();
  pro.listProc(request,response);
  gotoPage("/addProcedimiento.jsp?index="+ind,request,response);
// Editar tratamiento
} else if ( action.equalsIgnoreCase("addtrata") ) {
   EditTratamiento trat = new EditTratamiento();
   trat.editTra(request,response);
  gotoPage("/addTratamiento.jsp",request,response);
 // Almacenar tratamiento y volver al listado de
 // pacientes
} else if ( action.equalsIgnoreCase("addtratamiento") ) {
  AddTratamiento trata = new AddTratamiento();
  trata.addTra(request,response);
  EditDemograficos dem = new EditDemograficos();
  dem.editDemo(request,response);
  gotoPage("/addDemograficos.jsp?index="+ind,request,response);
  // Listado de informes
} else if ( action.equalsIgnoreCase("listinf") ) {
  ListInformes informe = new ListInformes();
  informe.list(request,response);
  gotoPage("/listInformes.jsp",request,response);
// Busqueda de informes
```

```
        } else if ( action.equalsIgnoreCase("searchinf") ) {
          gotoPage("/searchInforme.jsp",request,response);
        } else if ( action.equalsIgnoreCase("attributesearch") ) {
          SearchInforme informe = new SearchInforme();
          informe.attributeSearch(request,response);
          gotoPage("/listInformes.jsp",request,response);
        } else if ( action.equalsIgnoreCase("xpathsearch") ) {
          SearchInforme informe = new SearchInforme();
          informe.xpathSearch(request,response);
          gotoPage("/listInformes.jsp",request,response);
        // Borrar un informe
        } else if ( action.equalsIgnoreCase("deleteinforme") ) {
          DeleteInforme informe = new DeleteInforme();
          informe.delete(request,response);
          ListInformes list = new ListInformes();
          list.list(request,response);
          gotoPage("/listInformes.jsp",request,response);
        } else if ( action.equalsIgnoreCase("showxml")) {
           ShowXML muestra = new ShowXML();
           muestra.showXml(request,response);
        } else gotoPage("/error.jsp",request,response);
      }
      else {
        gotoPage("/error.jsp",request,response);
      }
    } catch( Exception e ) {
      e.printStackTrace();
```

```
          // there's not much else we can do if the response is committed
          if (response.isCommitted())
             return;
        // Catch the exception and send the user to the error page
        if (e.getMessage() != null ) {
           response.sendRedirect("/InformeAlta/error.jsp?error="                    +
URLEncoder.encode(e.getMessage()) );
        }
        else {
           response.sendRedirect("/InformeAlta/error.jsp" );
        }
      }
    }
    // Treat a form-get as a post, seding user to doPost
    public void doGet(HttpServletRequest request, HttpServletResponse
response ) throws ServletException, IOException {
      doPost(request,response);
    }
    // Forward the user on to the pages specified in address
    private void gotoPage( String address, HttpServletRequest request,
HttpServletResponse response ) throws ServletException, IOException {
      RequestDispatcher                dispatcher                =
getServletContext().getRequestDispatcher(address);
      dispatcher.forward(request, response);
    }
}
```

6.2- JSP

addAntecedentes.jsp

```
<%/* addAntecedentes.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>

<%/* En esta página jsp tendremos el formulario referente a los Antecedentes del paciente
 * y un enlace a la página donde podemos rellenar las Patologías previas. Cada vez que
 * accedamos a esta página aparecerán los datos del paciente que hayamos seleccionado y
 * modificando los campos que aparecen podremos rellenar los campos, para un nuevo paciente
 * o modificarlos para un paciente con datos anteriores
 */%>

<%@ include file="header2.jsp" %>

<jsp:include page="InformeAlta.EditAntecedentes" />

<FORM ACTION="/InformeAlta/servlet/InformeAlta.Task?addantecedentes&index=<%= index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="ADDANTECEDENTES">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >
```

```
  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >
  <FONT SIZE="+2">
    <center>Añadir los antecedentes personales</center>
  </FONT>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='0' width='620'>
   <tr>
   <td width='300'>
   <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300'>
      <tr>
      <td>
      <table align='center' border='0' cellPadding='0' cellSpacing='0'
width='300'>
         <tr>
         <td>
         <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
            <tr>
            <td width='300' align='left'>
                  <font> Alergias a medicamentos:</font>
            </td>
            </tr>
            <tr>
            <td>
                  <textarea rows=3 cols=30
name='ALERGIAMEDICAMENTO'><%=request.getAttribute("Alergia")%></text
area>
            </td>
```

```
            </tr>
          </table>
          </td>
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   <td width='20'> </td>
   <td width='300'>
   <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300'>
       <tr>
       <td>
       <table align='center' border='0' cellPadding='0' cellSpacing='0'
width='300''>
          <tr>
          <td>
          <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
             <tr>
             <td width='300' align='left'>
                <font> Alergias a otros:</font>
             </td>
             </tr>
             <tr>
             <td>
```

```
                <textarea rows=3 cols=30
name='ALERGIAOTROS'><%=request.getAttribute("AlergiaOtros")%></textare
a>

                </td>

                </tr>

            </table>

            </td>

            </tr>

        </table>

        </td>

        </tr>

    </table>

    </td>

    </tr>

</table>

</td>

</tr>

<tr>

<td> </td>

</tr>

<tr>

<td>

<table align='center' border='0' cellPadding='0' cellSpacing='0' width='620'>

    <tr>

    <td width='300'>

    <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300'>

        <tr>

        <td>
```

```
<table align='center' border='0' cellPadding='0' cellSpacing='0' width='300'>

<tr>

<td>

<table align='center' border='0' cellPadding='1' cellSpacing='2' width='300'>

<tr>

<td width='160' align='right'>

<font >Fumador: </font>

</td>

<td width='140'>

<select name='FUMADOR' >

<option value='<%=request.getAttribute("Fumador")%>'Selected><%=request.getAttribute("Fumador")%>

<option value='SI'>SI

<option value='NO'>NO

</select>

</td>

</tr>

<tr>

<td width='160' align='right' >

<font>Bebedor: </font>

</td>

<td width='140'>

<select name='BEBEDOR' >

<option value='<%=request.getAttribute("Bebedor")%>'Selected><%=request.getAttribute("Bebedor")%>

<option value='SI'>SI
```

```
                <option value='NO'>NO
                </select>
              </td>
              </tr>
            </table>
            </td>
            </tr>
        </table>
        </td>
        </tr>
    </table>
    </td>
    <td witdh='20'> </td>
    <td width='300' valign='top'>
    <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300'>
        <tr>
        <td>
        <table align='center' border='0' cellPadding='0' cellSpacing='0'
width='300'>
            <tr>
            <td>
            <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
              <tr>
              <td width='160' align='right'>
                <font >Diálisis: </font>
              </td>
              <td width='140'>
```

```
            <select name='DIALISIS' >

              <option
value='<%=request.getAttribute("Dialisis")%>'Selected><%=request.getAttribute
("Dialisis")%>

              <option value='SI'>SI

              <option value='NO'>NO

              </select>

            </td>

            </tr>

            <tr>

            <td width='160' align='right'>

              <font >Procedencia Diálisis: </font>

            </td>

            <td width='140'>

              <input type='text' name='PROCDIALISIS'
VALUE="<%=request.getAttribute("ProcDialisis")%>"  >

            </td>

            </tr>

          </table>

          </td>

          </tr>

       </table>

       </td>

       </tr>

    </table>

    </td>

    </tr>

</table>

</td>
```

```
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='0' width='620'>
   <tr>
   <td width='300' valign='top'>
   <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300'>
      <tr>
      <td>
      <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300'>
         <tr>
         <td>
         <table align='center' border='0' cellPadding='1' cellSpacing='2' width='300'>
            <tr>
            <td width='300' align='left' >
               <font> Otros Habitos Tóxicos:</font>
            </td>
            </tr>
            <tr>
            <td width='300'>
               <textarea rows=2 cols=30 name='OTROSHABITOS'><%=request.getAttribute("OtrosHabitos")%></textarea>
            </td>
```

```
            </tr>
          </table>
          </td>
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   <td width='20'> </td>
   <td width='300'>
   <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300'>
       <tr>
       <td>
       <table align='center' border='0' cellPadding='0' cellSpacing='0'
width='300'>
          <tr>
          <td>
          <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
             <tr>
             <td width='160' align='right'>
                <font >Diabetes Mellitus: </font>
             </td>
             <td width='140'>
              <select name='DIABETES' >
                 <option
value='<%=request.getAttribute("Diabetes")%>'Selected><%=request.getAttribu
te("Mellitus")%>
```

```
                <option value='SI'>SI
                <option value='NO'>NO
                </select>
              </td>
              </tr>
              <tr>
              <td width='160' align='right'>
                <font>HTA: </font>
              </td>
              <td width='140'>
               <select name='HTA' >
                <option
value='<%=request.getAttribute("HTA")%>'Selected><%=request.getAttribute("
HTA")%>
                <option value='SI'>SI
                <option value='NO'>NO
                </select>
              </td>
              </tr>
              <tr>
              <td width='160' align='right'>
                <font>Serología Viral: </font>
              </td>
              <td width='140'>
                <select  name='SEROLOGIA' >
                <option
value='<%=request.getAttribute("Serologia")%>'Selected><%=request.getAttrib
ute("Serologia")%>
                <option value='NINGUNA'>NINGUNA
```

```
                <option value='VIH'>VIH
                <option value='VHB'>VHB
                <option value='VHC'>VHC
                </select>
              </td>
              </tr>
            </table>
            </td>
            </tr>
          </table>
          </td>
          </tr>
        </table>
        </td>
        </tr>
      </table>
      </td>
      </tr>
    </table>
    </td>
    </tr>
  </table>
  </td>
  </tr>
  <tr>
  <td>
  <table align='center' border='0' cellPadding='0' cellSpacing='2' width='610'>
    <tr>
    <td>
    <table align='center' border='0' cellPadding='0' cellSpacing='0' width='610'>
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='610'>
```

```
          <tr>
          <td width='610' align='left' >
             <font > Antecedentes familiares:</font>
          </td>
          </tr>
          <tr>
          <td>
             <textarea rowls=7 cols=80 name='ANTFAMILIARES'
><%=request.getAttribute("AntFamiliares")%></textarea>
          </td>
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td > </td>
</tr><br><br>
<tr><center>
   <td align='center'>
      <input border='0' type='submit' name='addantecedentes&index=<%= index
%>' value='Anotar datos'>
   </td>
```

```
    <td align='center'>
        <input border='0' type='reset'>
    </td></center>
</tr>
</form>
<tr>
<td>
    <br>
    <hr >
    <br>
</td>
</tr>
<tr>
 <td width='300' align='left'>
   <font> Patologias previas</font>
 </td>
</tr>
<FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addpat&index=<%=
index %>" METHOD="POST">
   <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >
   <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >
<center>
 <INPUT border="0" TYPE="SUBMIT" NAME="addpat&index=<%= index %>"
VALUE="Patologias previas">
</center>
</FORM>
<tr>
```

```
<td width='750' height='20'> </td>

</tr>

<!-- Pie del documento -->

<%@ include file="footer.jsp" %>
```

addBioquimica.jsp

```
<%/* addBioquimica.jsp

 *

 * Proyecto Fin de Carrera 2002/2003

 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA

 * Y BASE DE DATOS XML NATIVAS

 *

 * @author  Irene Lavado Gomez

 */%>

<%/* En esta página jsp tendremos el formulario referente a las pruebas Bioquimicas

 * del paciente.  Cada vez que accedamos a esta página aparecerán los datos del paciente que

 * hayamos seleccionado y modificando los campos que aparecen podremos rellenar los campos,

 * para un nuevo paciente o modificarlos para un paciente con datos anteriores

 */%>

<%@ include file="header3.jsp" %>

<jsp:include page="InformeAlta.EditBioquimica" />

<FORM  ACTION="/InformeAlta/servlet/InformeAlta.Task" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="ADDBIOQUIMICA">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

  <FONT SIZE="+2">
```

```
    <center>Pruebas bioquimicas</center>
  </FONT>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
   <tr>
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
         <tr>
         <td width='132' align='right' >
            <font >Fecha de realizacion: </font>
         </td>
         <td width='220' align='left' >
            <input type='text' name='FECHABIOQUIM'
VALUE="<%=request.getAttribute("Fecha")%>" maxlength='30'
         </td>
         </tr>
         </table>
      </td>
      </tr>
      </table>
   </td>
   </tr>
   </table>
```

```
</td>
</tr>
<tr>
<td width='750' > </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='670' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='0' cellSpacing='0' width='670'
>
         <tr>
         <td>
            <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='670'>
            <tr>
               <td width='180' align='right' >
                     <font >Glucosa-s:</font>
               </td>
               <td width='30' align='left' >
                  <input type='text' name='GLUCOSA'
VALUE="<%=request.getAttribute("Glucosa")%>" maxlength='5' >
               </td>
               <td width='50' align='left' >
                  <font >mg/dl</font>
               </td>
               <td width='35' align='center'>
                  <font >70</font>
```

```
                </td>
                <td width='35' align='center' >
                    <font >110</font>
                </td>
                <td width='2' align='center' >
                </td>
                <td width='180' align='right' >
                    <font >GPT (ALT): </font>
                </td>
                <td width='30' align='left' >
                    <input type='text' name='GPT'
VALUE="<%=request.getAttribute("GPT")%>" maxlength='5' >
                </td>
                <td width='50' align='left'>
                    <font>U/L</font>
                </td>
                <td width='35' align='center' >
                    <font >0</font>
                </td>
                <td width='35' align='center' >
                    <font >40</font>
                </td>
               </tr>
               <tr>
                <td width='180' align='right' >
                <font >Urea-s: </font>
                </td>
                <td width='30' align='left' >
```

```
                <input type='text' name='UREA'
VALUE="<%=request.getAttribute("Urea")%>" maxlength='5' >
            </td>
            <td width='50' align='left'>
                <font >mg/dl</font>
            </td>
            <td width='35' align='center'>
                <font >20</font>
            </td>
            <td width='35' align='center' >
                <font >50</font>
            </td>
            <td width='2' align='center' >
            </td>
            <td width='180' align='right' >
                <font >GGT: </font>
            </td>
            <td width='30' align='left' >
                <input type='text' name='GGT'
VALUE="<%=request.getAttribute("GGT")%>" maxlength='5' >
            </td>
            <td width='50' align='left'>
                <font>U/L</font>
            </td>
            <td width='35' align='center' >
                <font >11</font>
            </td>
            <td width='35' align='center' >
```

```
                <font >49</font>
              </td>
             </tr>
             <tr>
               <td width='180' align='right' >
                 <font >Creatinina-s: </font>
              </td>
              <td width='30' align='left' >
                 <input type='text' name='CREATININA'
VALUE="<%=request.getAttribute("Creatinina")%>" maxlength='5' >
              </td>
              <td width='50' align='left' >
                 <font >mg/dl</font>
              </td>
              <td width='35' align='center' >
                 <font >0.70</font>
              </td>
              <td width='35' align='center' >
                 <font >1.50</font>
              </td>
              <td width='2' align='center' >
              </td>
              <td width='180' align='right' >
                 <font >Bilirrubina total-s: </font>
              </td>
              <td width='30' align='left' >
                 <input type='text' name='BILIRRUBINATOTAL'
VALUE="<%=request.getAttribute("BilirrubinaTotal")%>" maxlength='5' >
```

```
            </td>
            <td width='50' align='left'>
               <font >mg/dl</font>
            </td>
            <td width='35' align='center' >
               <font >0.0</font>
            </td>
            <td width='35' align='center' >
               <font >1.0</font>
            </td>
         </tr>
         <tr>
            <td width='180' align='right' >
               <font >Ácido úrico-s: </font>
            </td>
            <td width='30' align='left' >
               <input type='text' name='ACIDOURICO'
VALUE="<%=request.getAttribute("AcidoUrico")%>" maxlength='5' >
            </td>
            <td width='50' align='left' >
               <font >mg/dl</font>
            </td>
            <td width='35' align='center' >
               <font >2.6</font>
            </td>
            <td width='35' align='center' >
               <font >7.2</font>
            </td>
```

```
                <td width='2' align='center' >
                </td>
                <td width='180' align='right' >
                   <font >Bilirrubina directa-s: </font>
                </td>
                <td width='30' align='left' >
                   <input type='text' name='BILIRRUBINADIRECTA'
VALUE="<%=request.getAttribute("BilirrubinaDirecta")%>" maxlength='5' >
                </td>
                <td width='50' align='left' >
                   <font >mg/dl</font>
                </td>
                <td width='35' align='center' >
                   <font >0.0</font>
                </td>
                <td width='35' align='center'>
                   <font >0.3</font>
                </td>
             </tr>
             <tr>
                <td width='180' align='right' >
                   <font >Colesterol-s: </font>
                </td>
                <td width='30' align='left' >
                   <input type='text' name='COLESTEROL'
VALUE="<%=request.getAttribute("Colesterol")%>" maxlength='5' >
                </td>
                <td width='50' align='left'>
```

```
                <font >mg/dl</font>
            </td>
            <td width='35' align='center' >
                <font >150</font>
            </td>
            <td width='35' align='center' >
                <font>220</font>
            </td>
            <td width='2' align='center' >
            </td>
            <td width='180' align='right' >
                <font >Fosfatasa alcalina: </font>
            </td>
            <td width='30' align='left' >
                <input type='text' name='FOSFATASA'
VALUE="<%=request.getAttribute("Fosfatasa")%>" maxlength='5' >
            </td>
            <td width='50' align='left' >
                <font >U/L</font>
            </td>
            <td width='35' align='center'>
                <font >90</font>
            </td>
            <td width='35' align='center' >
                <font >258</font>
            </td>
        </tr>
        <tr>
```

```
                <td width='180' align='right' >
                   <font >Triglicéridos-s: </font>
                </td>
                <td width='30' align='left' >
                   <input type='text' name='TRIGLICERIDOS'
VALUE="<%=request.getAttribute("Trigliceridos")%>" maxlength='5' >
                </td>
                <td width='50' align='left' >
                   <font >mg/dl</font>
                </td>
                <td width='35' align='center' >
                   <font >70</font>
                </td>
                <td width='35' align='center' >
                   <font >170</font>
                </td>
                <td width='2' align='center' >
                </td>
                <td width='180' align='right' >
                   <font >Láctico deshidrogenasa:</font>
                </td>
                <td width='30' align='left'>
                   <input type='text' name='LACTICO'
VALUE="<%=request.getAttribute("Lactico")%>" maxlength='5' >
                </td>
                <td width='50' align='left' >
                   <font >U/L</font>
                </td>
```

```
                <td width='35' align='center' >
                    <font >230</font>
                </td>
                <td width='35' align='center' >
                    <font >460</font>
                </td>
            </tr>
            <tr>
                <td width='180' align='right' >
                    <font >CPK-s: </font>
                </td>
                <td width='30' align='left' >
                    <input type='text' name='CPK'
VALUE="<%=request.getAttribute("CPK")%>" maxlength='5' >
                </td>
                <td width='50' align='left' >
                    <font >U/L</font>
                </td>
                <td width='35' align='center' >
                    <font >10</font>
                </td>
                <td width='35' align='center' >
                    <font >195</font>
                </td>
                <td width='2' align='center' >
                </td>
                <td width='180' align='right' >
                    <font >Amilasa: </font>
```

```
                </td>
                <td width='30' align='left' >
                   <input type='text' name='AMILASA'
VALUE="<%=request.getAttribute("Amilasa")%>" maxlength='5' >
                </td>
                <td width='50' align='left' >
                   <font >U/L</font>
                </td>
                <td width='35' align='center' >
                   <font >0</font>
                </td>
                <td width='35' align='center' >
                   <font >220</font>
                </td>
             </tr>
             <tr>
                <td width='180' align='right' >
                   <font >Proteína total-s: </font>
                </td>
                <td width='30' align='left' >
                   <input type='text' name='PROTEINATOTAL'
VALUE="<%=request.getAttribute("ProteinaTotal")%>" maxlength='5' >
                </td>
                <td width='50' align='left' >
                   <font >g/dl</font>
                </td>
                <td width='35' align='center' >
                   <font >6.0</font>
```

```
                </td>
                <td width='35' align='center' >
                    <font >8.0</font>
                </td>
                <td width='2' align='center' >
                </td>
                <td width='180' align='right' >
                    <font >Sodio-s: </font>
                </td>
                <td width='30' align='left' >
                    <input type='text' name='SODIO'
VALUE="<%=request.getAttribute("Sodio")%>" maxlength='5' >
                </td>
                <td width='50' align='left' >
                    <font >mEq/L</font>
                </td>
                <td width='35' align='center' >
                    <font >135</font>
                </td>
                <td width='35' align='center' >
                    <font >145</font>
                </td>
            </tr>
            <tr>
                <td width='180' align='right' >
                    <font >Albúmina-s: </font>
                </td>
                <td width='30' align='left' >
```

```
                <input type='text' name='ALBUMINA'
VALUE="<%=request.getAttribute("Albumina")%>" maxlength='5' >
            </td>
            <td width='50' align='left' >
                <font >g/dl</font>
            </td>
            <td width='35' align='center' >
                <font >3.5</font>
            </td>
            <td width='35' align='center' >
                <font >5.0</font>
            </td>
            <td width='2' align='center' >
            </td>
            <td width='180' align='right'>
                <font>Potasio-s: </font>
            </td>
            <td width='30' align='left' >
                <input type='text' name='POTASIO'
VALUE="<%=request.getAttribute("Potasio")%>" maxlength='5' >
            </td>
            <td width='50' align='left' >
                <font >mEq/L</font>
            </td>
            <td width='35' align='center' >
                <font>3.5</font>
            </td>
            <td width='35' align='center'>
```

```
                <font>5.0</font>
              </td>
            </tr>
            <tr>
              <td width='180' align='right' >
                <font >GOT (AST): </font>
              </td>
              <td width='30' align='left' >
                <input type='text' name='GOT'
VALUE="<%=request.getAttribute("GOT")%>" maxlength='5' >
              </td>
              <td width='50' align='left'>
                <font >U/L</font>
              </td>
              <td width='35' align='center' >
                <font >0</font>
              </td>
              <td width='35' align='center' >
                <font >37</font>
              </td>
              <td width='2' align='center'>
              </td>
              <td width='180' align='right' >
                <font >Fósforo-s: </font>
              </td>
              <td width='30' align='left'>
                <input type='text' name='FOSFORO'
VALUE="<%=request.getAttribute("Fosforo")%>" maxlength='5' >
```

```
        </td>
        <td width='50' align='left' >
            <font>mg/dl</font>
        </td>
        <td width='35' align='center'>
            <font>2.5</font>
        </td>
        <td width='35' align='center' >
            <font >4.5</font>
        </td>
    </tr>
    <tr>
        <td width='180' align='right' >
        </td>
        <td width='30' align='left'>
        </td>
        <td width='50' align='left' >
        </td>
        <td width='35' align='center' >
        </td>
        <td width='35' align='center'>
        </td>
        <td width='2' align='center' >
        </td>
        <td width='180' align='right'>
            <font >Calcio-s: </font>
        </td>
```

```
                <td width='30' align='left' >
                    <input type='text' name='CALCIO'
VALUE="<%=request.getAttribute("Calcio")%>" maxlength='5' >
                </td>
                <td width='50' align='left'>
                    <font >mg/dl</font>
                </td>
                <td width='35' align='center'>
                    <font>8.5</font>
                </td>
                <td width='35' align='center'>
                    <font >10.5</font>
                </td>
            </tr>
            </table>
        </td>
        </tr>
        </table>
    </td>
    </tr>
    </table>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td>
```

```
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
   <tr>
    <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300' '>
         <tr>
         <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='300'>
          <tr>
            <td width='300' align='left' >
             <font > Observaciones:</font>
          </td>
          </tr>
          <tr>
            <td>
              <textarea rows=3 cols=30 name='OBSERVBIOQUIM'><%=request.getAttribute("Observ")%></textarea>
          </td>
          </tr>
          </table>
         </td>
         </tr>
         </table>
   </td>
   </tr>
   </table>
</td>
</tr>
```

```
<tr>
<td><br><br></td>
</tr>
<center>
<tr>
<td align='center'>
   <input border='0' type='submit' name='addbioquimica&index=<%= index %>' value='Anotar datos'>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td align='center'>
   <input border='0' type='reset'  >
</td>
</tr>
<tr>
<td> </td>
</tr>
<tr>
<td align='center'>
   <input border='0' type='submit'  name='NoAnotar' value='Volver sin anotar'>
</td>
</tr>
</center>
</form>
```

```
<tr>
  <td height='20'> </td>
</tr>
<!-- Pie del documento -->
<%@ include file="footer.jsp" %>
```

addCircunstancias.jsp

```
<%/* addCircunstancias.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%/* En esta página jsp tendremos el formulario referente a las Circunstancias de ingreso
 * del paciente.  Cada vez que accedamos a esta página aparecerán los datos del paciente que
 * hayamos seleccionado y modificando los campos que aparecen podremos rellenar los campos,
 * para un nuevo paciente o modificarlos para un paciente con datos anteriores
 */%>
<%@ include file="header2.jsp" %>
<jsp:include page="InformeAlta.EditCircunstancias" />
<FORM ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addcircunstanciasdom&index=<%= index %>" METHOD="POST">
  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="addcircunstanciasdom">
```

```
  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

  <FONT SIZE="+2">

    <center>Añadir circunstancias del ingreso</center>

  </FONT>

<tr>

<td>

<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360' >

   <tr>

    <td>

   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360'>

      <tr>

      <td>

      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='360'>

         <tr>

         <td width='132' align='right' >

            <font >Fecha de ingreso: </font>

         </td>

         <td width='220' align='left' >

            <input type='text' name='FECHAADMISION' VALUE="<%=request.getAttribute("Fecha")%>" maxlength='30'

         </td>

         </tr>

      </table>

      </td>

      </tr>

   </table>
```

```
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360' >
   <tr>
    <td>
      <table align='center' border='0' cellPadding='0' cellSpacing='0'
width='360'>
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='360'>
         <tr>
         <td width='140' align='right'>
            <font >Modo de ingreso: </font>
         </td>
         <td width='220' align='left'>
            <select name='MODOADMISION'>
               <option value='<%=request.getAttribute("Modo")%>'
Selected><%=request.getAttribute("Modo")%>
               <option value='Rutinario'>Rutinario
               <option value='Urgente'>Urgente
```

```
                <option value='De emergencia'>De emergencia
                <option value='Otros'>Otros
              </select>
            </td>
            </tr>
            <tr>
            <td width='140' align='right'>
              <font >Espec. admisora: </font>
            </td>
            <td width='220' align='left'>
              <input type='text'  name='ESPECIALIDADADMISORA' VALUE="<%=
request.getAttribute("Especialidad")%>" maxlength='30'
            </td>
            </tr>
          </table>
          </td>
          </tr>
      </table>
      </td>
      </tr>
</table>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td>
```

```
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360'>
   <tr>
    <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360'>
      <tr>
      <td>
      <table align='center' border='0' cellPadding='0' cellSpacing='0'
width='300'>
           <tr>
          <td>
            <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='360'>
              <tr>
              <td width='300' align='left' >
                 <font> Motivo del ingreso:</font>
              </td>
             </tr>
             <tr>
               <td>
                <textarea rows=3 cols=50
name='MOTIVOINGRESO'><%=request.getAttribute("Motivo")%></textarea>
             </td>
             </tr>
          </table>
          </td>
            </tr>
      </table>
      </td>
      </tr>
```

```
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td><br><br></td>
</tr>
<tr> <center>
<td align='center'>
   <input    border='0'    type='submit'name='addcircunstanciasdom&index=<%=
index %>' value='Anotar datos'>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td align='center'>
   <input border='0' type='reset'>
</td>  </center>
</tr>
</form>
<tr>
  <td height='20'> </td>
</tr>
<!-- Pie del documento -->
```

```
<%@ include file="footer.jsp" %>
```

addDemograficos.jsp

```
<%/* addDemograficos.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>

<%/* En esta página jsp tendremos el formulario referente a las Datos Demograficos
 * del paciente.  Cada vez que accedamos a esta página aparecerán los datos del paciente que
 * hayamos seleccionado y modificando los campos que aparecen podremos rellenar los campos,
 * para un nuevo paciente o modificarlos para un paciente con datos anteriores
 */%>

<%@ include file="header2.jsp" %>

<jsp:include page="InformeAlta.EditDemograficos" />

<FORM ACTION="/InformeAlta/servlet/InformeAlta.Task?action=adddemo&index=<%= index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="adddemo">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

   <FONT SIZE="+2">

     <center>Datos demograficos del paciente</center>
```

```
  </FONT>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360' '>
   <tr>
    <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='360'>
          <tr>
          <td width='140' align='right' >
             <font >D.N.I.: </font>
          </td>
          <td width='220' align='left' >
             <input type='text' disabled VALUE="<%= informe.getValorId() %>" name='VALORID' maxlength='10'>
           </td>
          </tr>
          <tr>
          <td width='140' align='right' >
             <font >Nº de historia: </font>
          </td>
          <td width='220' align='left' >
             <input type="text" VALUE="<%=request.getAttribute("ValorIdhc")%>" name='VALORIDHC' maxlength='20'>
          </td>
          </tr>
```

```
          <tr>
          <td width='140' align='right' >
             <font >Número S.S./T.S.: </font>
          </td>
          <td width='220' align='left' >
             <input  type='text'  VALUE="<%=request.getAttribute("ValorIdss")%>"
name='VALORIDSS' SIZE='12'>
          </td>
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td width='750'> </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360'>
   <tr>
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360'>
      <tr>
```

```
<td>
<table align='center' border='0' cellPadding='1' cellSpacing='2' width='360'>
<tr>
<td width='140' align='right'>
<font>Nombre: </font>
</td>
<td width='220' align='left'>
<input type='text' disabled VALUE="<%= informe.getNombre() %>" name='NOMBRE' maxlength='25' >
</td>
</tr>
<tr>
<td width='140' align='right'>
<font>Primer apellido: </font>
</td>
<td width='220' align='left'>
<input type='text' disabled VALUE="<%= informe.getPrimerApellido() %>" name="PRIMERAPELLIDO" maxlength='25'>
</td>
</tr>
<tr>
<td width='140' align='right'>
<font>Segundo apellido: </font>
</td>
<td width='220' align='left'>
<input type='text' disabled VALUE="<%= informe.getSegundoApellido() %>" name='SEGUNDOAPELLIDO' maxlength='25' >
```

```
          </td>
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td width='750'> </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360'>
   <tr>
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360'>
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='360'>
         <tr>
         <td width='132' align='right'>
            <font>Fecha de nacim.: </font>
         </td>
```

```
          <td width='220' align='left'>

            <input                                                    type='text'
VALUE="<%=request.getAttribute("FechaNacimiento")%>"
name='FECHANACIMIENTO' maxlength='10' >

          </td>

       </table>

       </td>

       </tr>

   </table>

   </td>

   </tr>

</table>

</td>

</tr>

<tr>

<td width='750'> </td>

</tr>

<tr>

<td>

<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360' >

   <tr>

   <td>

   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360' >

      <tr>

      <td>

      <table      align='center'      border='0'      cellPadding='1'      cellSpacing='2'
width='360'>

          <tr>

          <td width='140' align='right'>
```

```
            <font >Sexo: </font>
          </td>
          <td width='220' align='left'>
           <select name="SEXO">
                <option                value="<%=request.getAttribute("Sexo")%>"
Selected><%=request.getAttribute("Sexo")%>
                <option value="Hombre">Hombre
              <option value="Mujer">Mujer
                <option value="No especificado">No especificado
              <option value="Desconocido">Desconocido
           </select>
          </td>
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td width='750'> </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360'>
```

```
   <tr>
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360'>
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='360'>
          <tr>
          <td width='140' align='right'>
             <font>Tipo de dirección: </font>
            </td>
            <td width='220' align='left' >
          <select name="TIPODIRECCION">
                <option value="<%=request.getAttribute("TipoDireccion")%>" Selected><%=request.getAttribute("TipoDireccion")%>
                <option value="Casa">Casa
              <option value="Trabajo">Trabajo
                <option value="Temporal">Temporal
              <option value="Antigua">Antigua
                <option value="Otros">Otros
              <option value="Sin espificar">Sin especificar
           </select>
             </td>
          </tr>
          <tr>
          <td width='140' align='right'>
             <font>Domicilio: </font>
          </td>
```

```
<td width='220' align='left'>

<input type='text' VALUE="<%=request.getAttribute("Calle")%>"
name='CALLE' maxlength='45' >

</td>

</tr>

<tr>

<td width='140' align='right'>

<font>Número</font>

<font>- piso: </font>

</td>

<td width='120' align='left'>

<table align='center' border='0' cellPadding='0' cellSpacing='2'
width='120'>

<tr>

<td align='right'>

<input type='text' VALUE="<%=request.getAttribute("Numero")%>"
name='NUMERO' maxlength='4' >

</td>

<td width='20'> - </td>

<td align='right'>

<input type='text' VALUE="<%=request.getAttribute("Piso")%>"
name='PISO' maxlength='15'>

</td>

</tr>

</table>

</td>

</tr>

<tr>

<td width='140' align='right'>
```

```
            <font >Localidad: </font>
          </td>
          <td width='220' align='left'>
            <input  type='text'  VALUE="<%=request.getAttribute("Localidad")%>"
name='LOCALIDAD' maxlength='35' >
          </td>
          </tr>
          <tr>
          <td width='140' align='right'>
            <font >Provincia: </font>
          </td>
          <td width='220' align='left'>
            <input  type='text'  VALUE="<%=request.getAttribute("Provincia")%>"
name='PROVINCIA' maxlength='15' >
          </td>
          </tr>
          <tr>
          <td width='140' align='right'>
            <font>Código Postal: </font>
          </td>
          <td width='220' align='left'>
            <input                                                type='text'
VALUE="<%=request.getAttribute("CodigoPostal")%>"
name='CODIGOPOSTAL' maxlength='5' >
          </td>
          </tr>
        </table>
        </td>
        </tr>
```

```
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td width='750'> </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360'>


        <tr>
       <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='360'>
          <tr>
          <td width='140' align='right' >
             <font >Tipo Tfno (1): </font>
          </td>
            <td width='220' align='left' >
          <select name="TIPO1">
                <option value="<%=request.getAttribute("tipo1")%>"
Selected><%=request.getAttribute("tipo1")%>
                <option value="Casa">Casa
              <option value="Trabajo">Trabajo
                <option value="No especificado">No especificado
```

```
            <option value="Desconocido">Desconocido
          </select>
            </td>
          </tr>
          <tr>
          <td width='140' align='right' >
             <font >Clase Tfno: </font>
          </td>
           <td width='220' align='left' >
           <select name="CLASE1">
                <option                value="<%=request.getAttribute("clase1")%>"
Selected><%=request.getAttribute("clase1")%>
                <option value="Fijo">Fijo
              <option value="Movil">Movil
           </select>
          </td>
          </tr>
          <tr>
          <td width='140' align='right' >
             <font >Telefono: </font>
          </td>
          <td width='220' align='left' >
             <input      type='text'      VALUE="<%=request.getAttribute("tfno1")%>"
name='TFNO1' maxlength='25' >
          </td>
          </tr>
       </table>
       </td>
```

```
      </tr>
   </table>
   </td>
   </tr>
       <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='360'>
          <tr>
          <td width='140' align='right' >
             <font >Tipo Tfno (2): </font>
          </td>
            <td width='220' align='left' >
          <select name="TIPO2">
                <option value="<%=request.getAttribute("tipo2")%>" Selected><%=request.getAttribute("tipo2")%>
                <option value="Casa">Casa
             <option value="Trabajo">Trabajo
                <option value="No especificado">No especificado
             <option value="Desconocido">Desconocido
          </select>
          </td>
          </tr>
          <tr>
          <td width='140' align='right' >
             <font >Clase Tfno: </font>
          </td>
          <td width='220' align='left' >
```

```
          <select name="CLASE2">

                <option                value="<%=request.getAttribute("clase2")%>"
Selected><%=request.getAttribute("clase2")%>

                <option value="Fijo">Fijo

              <option value="Movil">Movil

          </select>

          </td>

          </tr>

          <tr>

          <td width='140' align='right' >

             <font >Telefono: </font>

          </td>

          <td width='220' align='left' >

             <input  type='text'      VALUE="<%=request.getAttribute("tfno2")%>"
name='TFNO2' maxlength='25' >

          </td>

          </tr>

       </table>

       </td>

       </tr>

    </table>

    </td>

    </tr>

        <tr>

       <td>

       <table  align='center'  border='0'  cellPadding='1'  cellSpacing='2'
width='360'>

          <tr>

          <td width='140' align='right'>
```

```
            <font>Tipo Tfno (3): </font>
          </td>
            <td width='220' align='left' >
             <select name="TIPO3">
                <option                value="<%=request.getAttribute("tipo3")%>"
Selected><%=request.getAttribute("tipo3")%>
                <option value="Casa">Casa
             <option value="Trabajo">Trabajo
                <option value="No especificado">No especificado
             <option value="Desconocido">Desconocido
           </select>
            </td>
          </tr>
          <tr>
          <td width='140' align='right' >
             <font >Clase Tfno: </font>
          </td>
           <td width='220' align='left' >
           <select name="CLASE3">
                <option                value="<%=request.getAttribute("clase3")%>"
Selected><%=request.getAttribute("clase3")%>
                <option value="Fijo">Fijo
             <option value="Movil">Movil
           </select>
             </td>
          </tr>
          <tr>
          <td width='140' align='right' >
```

```
            <font >Telefono: </font>
          </td>
          <td width='220' align='left' >
            <input type='text' VALUE="<%=request.getAttribute("tfno3")%>"
name='TFNO3' maxlength='25' >
          </td>
          </tr>
        </table>
        </td>
        </tr>
    </table>
    </td>
    </tr>
</table>
</td>
</tr>
<tr>
<td ><br><br></td>
</tr>
<tr> <center>
<td align='center'>
    <INPUT border="0" TYPE="SUBMIT" name='adddemo&index=<%= index
%>' VALUE="Salvar datos">
</td>
</tr>
<tr>
<td > </td>
</tr>
```

```
<tr>
<td align='center'>
    <input border='0' type='reset'>
</td> </center>
</tr>
<tr>
  <td height='20'> </td>
</tr>
</form>
<!-- Pie del documento -->
<%@ include file="footer.jsp" %>
```

addExploracion.jsp

```
<%/* addExploracion.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%/* En esta página jsp tendremos el formulario referente a la Exploracion Fisica
 * del paciente.  Cada vez que accedamos a esta página aparecerán los datos del paciente que
 * hayamos seleccionado y modificando los campos que aparecen podremos rellenar los campos,
 * para un nuevo paciente o modificarlos para un paciente con datos anteriores
 */%>
```

```
<%@ include file="header3.jsp" %>

<jsp:include page="InformeAlta.EditExploracion" />

<FORM ACTION="/InformeAlta/servlet/InformeAlta.Task?addexploracion" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="ADDEXPLORACION">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

  <FONT SIZE="+2">

    <center>Exploracion Fisica</center>

  </FONT>

<tr>

<td>

<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >

   <tr>

   <td>

   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300' >

      <tr>

      <td>

      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='300'>

         <tr>

         <td width='132' align='right' >

            <font >Fecha de realizacion: </font>

         </td>

         <td width='220' align='left'>

            <input type='text' name='FECHAEXPLOR' VALUE="<%=request.getAttribute("Fecha")%>" maxlength='30'
```

```
          </td>
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td width='750' > </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
   <tr>
    <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300' >
       <tr>
       <td>
       <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
          <tr>
          <td width='300' align='left' >
             <font > Medida de signos vitales:</font>
          </td>
```

```
          </tr>
        </table>
          </td>
        </tr>
    </table>
    </td>
    </tr>
    <tr>
    <td>
    <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300' >
        <tr>
        <td>
        <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
          <tr>
          <td width='120' align='right' >
            <font >Peso (kg): </font>
          </td>
          <td width='30' align='left' >
            <input type='text' name='PESO'
VALUE="<%=request.getAttribute("Peso")%>" maxlength='3' >
          </td>
          <td width='120' align='right' >
            <font >Talla (cm): </font>
          </td>
          <td width='30' align='left' >
            <input type='text' name='TALLA'
VALUE="<%=request.getAttribute("Talla")%>" maxlength='3' >
          </td>
```

```
        </tr>
        <tr>
        <td width='120' align='right' >
           <font >FC (lpm): </font>
        </td>
        <td width='30' align='left' >
           <input type='text' name='FC'
VALUE="<%=request.getAttribute("FC")%>" maxlength='3' >
        </td>
        <td width='120' align='right' >
           <font >TA (mmHg): </font>
        </td>
        <td width='30' align='left' >
           <input type='text' name='TA'
VALUE="<%=request.getAttribute("TA")%>" maxlength='6' >
        </td>
        </tr>
        <tr>
        <td width='120' align='right'>
            <font >Tª axilar (ºC):</font>
        </td>
        <td width='30' align='left' >
           <input type='text' name='TEMP'
VALUE="<%=request.getAttribute("Temp")%>" maxlength='4' >
        </td>
        <td width='120' align='right' >
        </td>
        <td width='30' align='left' >
        </td>
```

```
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td> </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='0' width='620'>
   <tr>
   <td width='300'>
   <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
       <tr>
       <td>
       <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300'
>
          <tr>
          <td>
            <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
              <tr>
```

```
            <td width='300' align='left' >
               <font > Aspecto general:</font>
            </td>
            </tr>
            <tr>
            <td>
               <textarea rows=3 cols=30 name='ASPECTOGENERAL' ><%=request.getAttribute("Aspecto")%></textarea>
            </td>
            </tr>
         </table>
         </td>
         </tr>
      </table>
      </td>
      </tr>
   </table>
   </td>
   <td width='10'> </td>
   <td width='300'>
   <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300'>
         <tr>
         <td>
         <table align='center' border='0' cellPadding='1' cellSpacing='2' width='300'>
```

```
            <tr>
            <td width='300' align='left'
               <font > Cabeza y cuello:</font>
            </td>
            </tr>
            <tr>
            <td>
               <textarea rows=3 cols=30 name='CABEZA'
><%=request.getAttribute("Cabeza")%></textarea>
            </td>
            </tr>
          </table>
          </td>
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td>
```

```
<table align='center' border='0' cellPadding='0' cellSpacing='0' width='620'>
   <tr>
   <td width='300'>
   <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='0' cellSpacing='0'
width='300'>
         <tr>
         <td>
         <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
            <tr>
            <td width='300' align='left'>
               <font > Tórax:</font>
            </td>
            </tr>
            <tr>
            <td>
              <textarea rows=3 cols=30
name='TORAX'><%=request.getAttribute("Torax")%></textarea>
            </td>
            </tr>
         </table>
         </td>
         </tr>
      </table>
      </td>
      </tr>
```

```
   </table>
   </td>
   <td width='10'> </td>
   <td width='300'>
   <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300'
>
         <tr>
         <td>
         <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
            <tr>
            <td width='300' align='left' >
               <font > Abdomen:</font>
            </td>
            </tr>
            <tr>
            <td>
               <textarea rows=3 cols=30
name='ABDOMEN'><%=request.getAttribute("Abdomen")%></textarea>
            </td>
            </tr>
         </table>
         </td>
         </tr>
      </table>
      </td>
```

```
      </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='0' width='620'>
   <tr>
   <td width='300'>
   <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300'
>
         <tr>
         <td>
         <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
            <tr>
            <td width='300' align='left' >
               <font > Extremidades:</font>
            </td>
```

```
            </tr>
            <tr>
            <td>
              <textarea rows=3 cols=30
name='EXTREMIDADES'><%=request.getAttribute("Extremidades")%></textar
ea>
            </td>
            </tr>
          </table>
          </td>
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   <td width='10'> </td>
   <td width='300'>
   <table align='center' border='0' cellPadding='0' cellSpacing='2' width='300'>
      <tr>
      <td>
      <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300'
>
         <tr>
         <td>
         <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
            <tr>
            <td width='300' align='left'>
```

```
                <font > Otras observaciones:</font>
            </td>
            </tr>
            <tr>
            <td>
                <textarea rows=3 cols=30 name='OBSERVEXPLOR'
><%=request.getAttribute("Observ")%></textarea>
            </td>
            </tr>
          </table>
          </td>
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td ><br><br></td>
</tr>
<tr>
<td align='center'>
   <input border='0' type='submit' name='addexploracion&index=<%= index %>'
value='Anotar datos'>
```

```
</td>
</tr>
<tr>
<td> </td>
</tr>
<tr>
<td align='center'>
   <input border='0' type='reset'>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td align='center'>
   <input border='0' type='submit'name='NoAnotar' value='Volver sin anotar'>
</td>
</tr>
</form>
<tr>
  <td height='20'> </td>
</tr>
<!-- Pie del documento -->
<%@ include file="footer.jsp" %>
```

addHemograma.jsp

```
<%/* addHemograma.jsp
 *
```

```
 * Proyecto Fin de Carrera 2002/2003

 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA

 * Y BASE DE DATOS XML NATIVAS

 *

 * @author  Irene Lavado Gomez

 */%>

<%/* En esta página jsp tendremos el formulario referente a los datos del Hemograma

 * del paciente.  Cada vez que accedamos a esta página aparecerán los datos del paciente que

 * hayamos seleccionado y modificando los campos que aparecen podremos rellenar los campos,

 * para un nuevo paciente o modificarlos para un paciente con datos anteriores

 */%>

<%@ include file="header3.jsp" %>

<jsp:include page="InformeAlta.EditHemograma" />

<FORM  ACTION="/InformeAlta/servlet/InformeAlta.Task" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="ADDHEMOGRAMA">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >

   <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

 <FONT SIZE="+2">

    <center>Hemograma</center>

  </FONT>

<tr>

<td>

<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >

   <tr>

   <td>
```

```
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
         <tr>
         <td width='132' align='right' >
            <font >Fecha de realización: </font>
         </td>
         <td width='220' align='left' >
            <input type='text' name='FECHAHEMO'
VALUE="<%=request.getAttribute("Fecha")%>" maxlength='30'
         </td>
         </tr>
      </table>
      </td>
      </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td width='750' > </td>
</tr>
<tr>
<td>
```

```
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='650' >
   <tr>
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='650' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='650'>
            <tr>
               <td width='120' align='right' >
                  <font >Hematíes: </font>
               </td>
               <td width='30' align='left' >
                  <input type='text' name='HEMATIES' VALUE="<%=request.getAttribute("Hematies")%>" maxlength='5' >
               </td>
               <td width='90' align='left' >
                  <font >x1000^2/uL</font>
               </td>
               <td width='40' align='center' >
                  <font >4.60</font>
               </td>
               <td width='40' align='center' >
                  <font >5.70</font>
               </td>
               <td width='2' align='center' >
               </td>
               <td width='120' align='right' >
```

```
                <font >Leucocitos: </font>
            </td>
            <td width='30' align='left' >
                <input type='text' name='LEUCOCITOS'
VALUE="<%=request.getAttribute("Leucocitos")%>" maxlength='5' >
            </td>
            <td width='90' align='left' >
                <font >x1000/uL</font>
            </td>
            <td width='40' align='center' >
                <font >4.0</font>
            </td>
            <td width='40' align='center'>
                <font >11.0</font>
            </td>
        </tr>
        <tr>
            <td width='120' align='right' >
                <font >Hemoglobina: </font>
            </td>
            <td width='30' align='left' >
                <input type='text' name='HEMOGLOBINA'
VALUE="<%=request.getAttribute("Hemoglobina")%>" maxlength='5' >
            </td>
            <td width='90' align='left' >
                <font >g/dl</font>
            </td>
            <td width='40' align='center' >
```

```
                <font >13.0</font>
              </td>
              <td width='40' align='center' >
                <font >17.5</font>
              </td>
              <td width='2' align='center' >
              </td>
              <td width='120' align='right'>
                <font >Segmentados: </font>
              </td>
              <td width='30' align='left' >
                <input type='text' name='SEGMENTADOS'
VALUE="<%=request.getAttribute("Segmentados")%>" maxlength='5' >
              </td>
              <td width='90' align='left'>
                <font >%</font>
              </td>
              <td width='40' align='center' >
                <font >42.0</font>
              </td>
              <td width='40' align='center' >
                <font >75.0</font>
              </td>
            </tr>
            <tr>
              <td width='120' align='right' >
                <font>Hematocrito: </font>
              </td>
```

```
                <td width='30' align='left' >
                    <input type='text' name='HEMATOCRITO'
VALUE="<%=request.getAttribute("Hematocrito")%>" maxlength='5'>
                </td>
                <td width='90' align='left' >
                    <font >%</font>
                </td>
                <td width='40' align='center'>
                    <font >40.0</font>
                </td>
                <td width='40' align='center' >
                    <font >54.0</font>
                </td>
                <td width='2' align='center' >
                </td>
                <td width='120' align='right' >
                    <font >Linfocitos: </font>
                </td>
                <td width='30' align='left' >
                    <input type='text' name='LINFOCITOS'
VALUE="<%=request.getAttribute("Linfocitos")%>" maxlength='5'>
                </td>
                <td width='90' align='left'>
                    <font >%</font>
                </td>
                <td width='40' align='center' >
                    <font >16.0</font>
                </td>
```

```
                <td width='40' align='center' >
                    <font >45.0</font>
                </td>
            </tr>
            <tr>
                <td width='120' align='right' >
                    <font >V.C.M.: </font>
                </td>
                <td width='30' align='left' >
                    <input type='text' name='VCM'
VALUE="<%=request.getAttribute("VCM")%>" maxlength='5' >
                </td>
                <td width='90' align='left'>
                    <font >fl</font>
                </td>
                <td width='40' align='center' >
                    <font >82.0</font>
                </td>
                <td width='40' align='center' >
                    <font >98.0</font>
                </td>
                <td width='2' align='center' >
                </td>
                <td width='120' align='right' >
                    <font >Monocitos: </font>
                </td>
                <td width='30' align='left' >
```

```
                <input type='text' name='MONOCITOS'
VALUE="<%=request.getAttribute("Monocitos")%>" maxlength='5'>
            </td>
            <td width='90' align='left' >
                <font >%</font>
            </td>
            <td width='40' align='center'>
                <font >4.0</font>
            </td>
            <td width='40' align='center' >
                <font >12.0</font>
            </td>
        </tr>
        <tr>
            <td width='120' align='right' >
                <font >H.C.M.: </font>
            </td>
            <td width='30' align='left' >
                <input type='text' name='HCM'
VALUE="<%=request.getAttribute("HCM")%>" maxlength='5' >
            </td>
            <td width='90' align='left' >
                <font >pg</font>
            </td>
            <td width='40' align='center' >
                <font >27.0</font>
            </td>
            <td width='40' align='center' >
```

```
                <font >31.0</font>
            </td>
            <td width='2' align='center'>
            </td>
            <td width='120' align='right' >
                <font >Eosinófilos: </font>
            </td>
            <td width='30' align='left' >
                <input type='text' name='EOSINOFILOS'
VALUE="<%=request.getAttribute("Eosinofilos")%>"  maxlength='5' >
            </td>
            <td width='90' align='left' >
                <font >%</font>
            </td>
            <td width='40' align='center' >
                <font >1.0</font>
            </td>
            <td width='40' align='center' >
                <font >4.0</font>
            </td>
        </tr>
        <tr>
            <td width='120' align='right' >
                <font >C.H.C.M.: </font>
            </td>
            <td width='30' align='left' >
                <input type='text' name='CHCM'
VALUE="<%=request.getAttribute("CHCM")%>" maxlength='5'>
```

```
                </td>
                <td width='90' align='left' >
                   <font >g/dl</font>
                </td>
                <td width='40' align='center' >
                   <font >31.0</font>
                </td>
                <td width='40' align='center' >
                   <font >38.0</font>
                </td>
                <td width='2' align='center' >
                </td>
                <td width='120' align='right' >
                   <font >Basófilos: </font>
                </td>
                <td width='30' align='left' >
                   <input type='text' name='BASOFILOS'
VALUE="<%=request.getAttribute("Basofilos")%>" maxlength='5' >
                </td>
                <td width='90' align='left' >
                   <font >%</font>
                </td>
                <td width='40' align='center' >
                   <font >0.0</font>
                </td>
                <td width='40' align='center' >
                   <font >2.0</font>
                </td>
```

```
            </tr>
            <tr>
               <td width='120' align='right' >
                  <font >R.D.W.: </font>
               </td>
               <td width='30' align='left' >
                  <input type='text' name='RDW'
VALUE="<%=request.getAttribute("RDW")%>" maxlength='5' >
               </td>
               <td width='90' align='left' >
                  <font >%</font>
               </td>
               <td width='40' align='center' >
                  <font >11.0</font>
               </td>
               <td width='40' align='center' >
                  <font >15.0</font>
               </td>
               <td width='2' align='center' >
               </td>
               <td width='120' align='right' >
                  <font >Plaquetas: </font>
               </td>
               <td width='30' align='left' >
                  <input type='text' name='PLAQUETAS'
VALUE="<%=request.getAttribute("Plaquetas")%>" maxlength='5'>
               </td>
               <td width='90' align='left' >
```

```
                    <font >x1000/uL</font>
                </td>
                <td width='40' align='center' >
                    <font >140</font>
                </td>
                <td width='40' align='center' >
                    <font >450</font>
                </td>
            </tr>
        </table>
        </td>
        </tr>
    </table>
    </td>
    </tr>
</table>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
    <tr>
     <td>
    <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300' >
```

```
      <tr>
        <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='300'>
        <tr>
        <td width='300' align='left'>
           <font > Observaciones:</font>
        </td>
        </tr>
        <tr>
        <td>
           <textarea rows=3 cols=30 name='OBSERVHEMO'><%=request.getAttribute("ObservHemo")%></textarea>
        </td>
        </tr>
      </table>
      </td>
      </tr>
     </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td ><br><br></td>
</tr>
<tr>
```

```
<td align='center'>
   <input  border='0'  type='submit'    name='addhemograma&index=<%=  index
%>' value='Anotar datos'>
</td>
</tr>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td align='center'>
   <input border='0' type='reset'  >
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td align='center'>
   <input border='0' type='submit'  name='NoAnotar' value='Volver sin anotar'>
</td>
</tr>
</form>
<tr>
  <td height='20'> </td>
</tr>
<!-- Pie del documento -->
<%@ include file="footer.jsp" %>
```

addInforme.jsp

```
<%/* addInforme.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%/* En esta página jsp tendremos el formulario referente a la creación de un nuevo informe,
 * para ello tendremos que rellenar el Dni, nombre y apellidos, y al anotarlos pasaremos al
 * listado de pacientes, desde donde podremos seleccionarlo y seguir rellenando sus datos
 */%>
<%@ include file="header.jsp" %>
<jsp:useBean id="group" class="InformeAlta.Group"  scope="session" />
<FORM NAME="addinforme" ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addinforme" METHOD="POST">
  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="ADDINFORME">
   <FONT SIZE="+2">
     <center>Añadir un informe</center>
   </FONT>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360' >
    <tr>
```

```
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='360'>
         <tr>
         <td width='140' align='right'>
            <font >D.N.I.: </font>
         </td>
         <td width='220' align='left' >
            <input type='text'  name='VALORID' maxlength='10'>
         </td>
         </tr>
      </table>
      </td>
      </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td width='750' > </td>
</tr>
<tr>
<td>
```

```
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360' >
   <tr>
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='360'>
         <tr>
         <td width='140' align='right' >
            <font >Nombre: </font>
         </td>
         <td width='220' align='left' >
            <input type='text'  name='NOMBRE' maxlength='25' >
         </td>
         </tr>
         <tr>
         <td width='140' align='right' >
            <font >Primer apellido: </font>
         </td>
         <td width='220' align='left' >
            <input type='text'  name="PRIMERAPELLIDO" maxlength='25'>
         </td>
         </tr>
         <tr>
         <td width='140' align='right' >
            <font >Segundo apellido: </font>
         </td>
```

```
          <td width='220' align='left' >
             <input type='text'  name='SEGUNDOAPELLIDO' maxlength='25' >
          </td>
          </tr>
       </table>
       </td>
       </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td width='750' > </td>
</tr>
<CENTER>
<tr>
<td align='center'>
   <input type='hidden' name='NuePac' >
   <INPUT  border="0"  TYPE="SUBMIT"   name='addinforme'  VALUE="Salvar datos">
</td>
</tr>
<tr>
<td align='center'>
   <input border='0' type='reset' '>
</td>
```

```
</tr>
</CENTER>
</form>
<tr>
  <td height='20'> </td>
  <td height='20'> </td>
</tr>
<!-- Pie del documento -->
<%@ include file="footer.jsp" %>
```

addJuicio.jsp

```
<%/* addJuicio.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%/* En esta página jsp tendremos el formulario referente a los diagnosticos a los que el doctor
 * ha llegado.  Dentro de este formulario tendremos varias opciones, en primer lugar la
 * posibilidad de rellenar nuevos diagnosticos, en segundo lugar un listado de los diagnosticos
 * ya establecidos, y en tercer lugar la posibilidad de determinar la evolucion del paciente
 */%>
<%@ include file="header2.jsp" %>
    <FONT SIZE="+2">
```

```
    <center>Juicio Clínico</center>
  </FONT>
<FORM ACTION="/InformeAlta/servlet/InformeAlta.Task" METHOD="POST">
  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="ADDJUICIO">
  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >
  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >
<center>
<table width="80%" >
  <tr bgcolor="#8fbc8b"><th>Introducir un nuevo juicio</th></tr>
</table>
</center>
<center>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360' >
   <tr>
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='360'>
         <tr>
         <td width='140' align='right' >
            <font >Diagnostico: </font>
         </td>
         <td width='220' align='left'>
```

```
            <input type='text' name='NOMBREDIAGN' value=’null’
maxlength='25' >
          </td>
          </tr>
          <tr>
          <td width='140' align='right' >
             <font >Id diagnostico: </font>
          </td>
          <td width='220' align='left' >
             <input type='text' name="IDDIAGN" value=’null’ maxlength='25'>
          </td>
          </tr>
          <tr>
          <td width='140' align='right' >
             <font >Caracter: </font>
          </td>
          <td width='220' align='left' >
             <select name='CARACTER' value=’null’>
                <option value="">
                <option value='Principal'>Principal
                <option value='Secundario'>Secundario
             </select>
          </td>
          </tr>
       </table>
       </td>
       </tr>
    </table>
```

```
   </td>
   </tr>
</table>
</td>
</tr>
</center>
<tr>
<td > </td>
</tr>
<center>
<tr>
<td align='center'>
   <input border='0' type='submit' name='addjuicio&index=<%= index %>' value='Añadir diagnostico'>
</td>
</tr>
</center>
</form>
<center>
<table width="80%" border="1">
 <tr bgcolor="#8fbc8b"><th>Diagnostico</th><th>CIE</th><th>Caracter</th></tr>
</center>
<center>
        <tr align=center>
            <td align='center'>
                <font ><%= request.getAttribute("Nombre0") %></font>
            </td>
```

```
        <td align='center'>
            <font ><%=request.getAttribute("Id0")%></font>
        </td>
        <td align='center'>
          <font ><%=request.getAttribute("Caracter0")%></font>
        </td>
      </tr>
      <tr align=center>
        <td align='center'>
            <font ><%= request.getAttribute("Nombre1") %></font>
        </td>
        <td align='center'>
            <font ><%=request.getAttribute("Id1")%></font>
        </td>
        <td align='center'>
          <font ><%=request.getAttribute("Caracter1")%></font>
        </td>
      </tr>
     <tr align=center>
        <td align='center'>
            <font ><%= request.getAttribute("Nombre2") %></font>
        </td>
        <td align='center'>
            <font ><%=request.getAttribute("Id2")%></font>
        </td>
        <td align='center'>
          <font ><%=request.getAttribute("Caracter2")%></font>
```

```
            </td>
          </tr>
          <tr align=center>
            <td align='center'>
              <font ><%= request.getAttribute("Nombre3") %></font>
            </td>
            <td align='center'>
              <font ><%=request.getAttribute("Id3")%></font>
            </td>
            <td align='center'>
              <font ><%=request.getAttribute("Caracter3")%></font>
            </td>
          </tr>
</center>
</table>
<BR>
<tr>
  <td height='20'> </td>
</tr>
<FORM  ACTION="/InformeAlta/servlet/InformeAlta.Task" METHOD="POST">
   <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="addjuiciocom">
  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >
  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
    <tr>
```

```
 <td>
<table align='center' border='0' cellPadding='0' cellSpacing='0' width='300' >
  <tr>
  <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300'>
     <tr>
    <td>
    <table align='center' border='0' cellPadding='1' cellSpacing='2' width='300'>
      <tr>
      <td width='300' align='left' >
        <font > Evolución del paciente:</font>
      </td>
      </tr>
      <tr>
      <td>
        <textarea rows=3 cols=30 name='EVOLUCION' ></textarea>
      </td>
      </tr>
    </table>
    </td>
    </tr>
  </table>
   </td>
  </tr>
</table>
</td>
```

```
    </tr>
</table>
</td>
</tr>
<tr>
<td > </td>
</tr>
<br>
<center>
<tr>
<td align='center'>
    <input  border='0'  type='submit'  name='addjuiciocom&index=<%=  index  %>'
value='Anotar datos'>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td align='center'>
    <input border='0' type='reset' >
</td>
</tr>
</center>
</form>
<tr>
  <td height='20'> </td>
</tr>
```

```
<!-- Pie del documento -->

<%@ include file="footer.jsp" %>
```

addPatologias.jsp

```
<%/* addPatologias.jsp

 *

 * Proyecto Fin de Carrera 2002/2003

 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA

 * Y BASE DE DATOS XML NATIVAS

 *

 * @author  Irene Lavado Gomez

 */%>

<%/* En esta página jsp tendremos el formulario referente a las patologias previas que el

 * paciente ha sufrido.  Dentro de este formulario tendremos varias opciones, en primer lugar

 * la posibilidad de rellenar nuevas patologias,y  en segundo lugar un listado de patologias

 */%>

<%@ include file="header3.jsp" %>

<jsp:include page="InformeAlta.ListPatologias" />

  <FONT SIZE="+2">

     <center>Añadir patologias</center>

   </FONT>

<FORM ACTION="/InformeAlta/servlet/InformeAlta.Task" METHOD="POST">

   <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="ADDPATOLOGIAS">

   <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >

   <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

<center>
```

```
 <table width="80%">
   <tr bgcolor="#8fbc8b"><th>Introducir una nueva patologia previa</th></tr>
   </table>
</center>
<center>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360' >
    <tr>
    <td>
    <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360' >
        <tr>
        <td>
        <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='360'>
            <tr>
            <td width='140' align='right' >
                <font >Nombre Patologia: </font>
            </td>
            <td width='220' align='left'>
                <input type='text' name='NOMBREPATOLOGIA' value=’null’
maxlength='25' >
            </td>
            </tr>
            <tr>
            <td width='140' align='right' >
                <font >Id Patologia: </font>
            </td>
```

```
        <td width='220' align='left' >
          <input type='text' name="IDPATOLOGIA" value=’null’ maxlength='25'>
        </td>
        </tr>
        <tr>
        <td width='140' align='right' >
          <font >Fecha Patologia: </font>
        </td>
        <td width='220' align='left' >
          <input type='text' name='FECHAPATOLOGIA' value=’null’ maxlength='25' >
        </td>
        </tr>
      </table>
      </td>
      </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
</center>
 <tr>
<td > </td>
</tr>
<BR>
```

```
<center>

<tr>

<td align='center'>

   <input border='0' type='submit' name='addpatologias&index=<%= index %>' value='Añadir patología'>

</td>

</tr>

</center>

</form>

<FORM ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addcircunsdom&index=<%= index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >

<center>

 <table width="80%" BORDER="1">

  <tr bgcolor="#8fbc8b"><th>Nombre de la Patologia</th><th>CIE</th><th>Fecha Patologia</th></tr>

        <tr align=center>

           <td align='center'>

              <font ><%= request.getAttribute("Nombre0") %></font>

           </td>

           <td align='center'>

              <font ><%=request.getAttribute("Id0")%></font>

           </td>

           <td align='center'>

             <font ><%=request.getAttribute("Fecha0")%></font>

           </td>

        </tr>
```

```
        <tr align=center>
        <td align='center'>
            <font ><%= request.getAttribute("Nombre1") %></font>
        </td>
        <td align='center'>
            <font ><%=request.getAttribute("Id1")%></font>
        </td>
        <td align='center'>
          <font ><%=request.getAttribute("Fecha1")%></font>
        </td>
     </tr>
    <tr align=center>
        <td align='center'>
            <font ><%= request.getAttribute("Nombre2") %></font>
        </td>
        <td align='center'>
            <font ><%=request.getAttribute("Id2")%></font>
        </td>
        <td align='center'>
          <font ><%=request.getAttribute("Fecha2")%></font>
        </td>
     </tr>
    <tr align=center>
        <td align='center'>
            <font ><%= request.getAttribute("Nombre3") %></font>
        </td>
        <td align='center'>
```

```
                <font ><%=request.getAttribute("Id3")%></font>
            </td>
            <td align='center'>
              <font ><%=request.getAttribute("Fecha3")%></font>
            </td>
          </tr>
   </table>
</center>
<BR>
<tr>
  <td height='20'> </td>
</tr>
<tr>
  <td height='20'> </td>
</tr>
<center>
 <INPUT border="0" TYPE="SUBMIT" NAME="addcircunsdom&index=<%= index %>" VALUE="Continuar">
</center>
</FORM><BR>
<tr>
  <td height='20'> </td>
</tr>
<!-- Pie del documento -->
<%@ include file="footer.jsp" %>
```

addProcedimiento.jsp

```
<%/* addProcedimiento.jsp
```

```
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%/* En esta página jsp tendremos el formulario referente a los procedimientos que se han
 * llevado a cabo durante la hospitalización del paciente. Dentro de este formulario tendremos
 * varias opciones, en primer lugar la posibilidad de rellenar nuevos procedimientos
 * y en segundo lugar un listado de los procedimientos ya realizados
 */%>
<%@ include file="header2.jsp" %>
    <FONT SIZE="+2">
     <center>Procedimiento realizado</center>
  </FONT>
<FORM ACTION="/InformeAlta/servlet/InformeAlta.Task" METHOD="POST">
  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="ADDPROCEDIMIENTO">
  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >
  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >
<center>
 <table width="80%">
  <tr bgcolor="#8fbc8b"><th>Introducir procedimientos realizados</th></tr>
  </table>
```

```
</center>
<center>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360' >
   <tr>
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='360'>
         <tr>
         <td width='140' align='right' >
            <font >Procedimiento: </font>
         </td>
         <td width='220' align='left'>
            <input type='text' name='NOMBREPROCED' value=’null’ maxlength='25' >
         </td>
         </tr>
         <tr>
         <td width='140' align='right' >
            <font >Id procedimiento: </font>
         </td>
         <td width='220' align='left' >
            <input type='text' name="IDPROCED" value=’null’ maxlength='25'>
         </td>
```

```
          </tr>
          <tr>
          <td width='140' align='right' >
             <font >Comentario: </font>
          </td>
          <td width='220' align='left' >
                <textarea name='OBSERVPROCED' value='null' ></textarea>
          </td>
          </tr>
       </table>
       </td>
       </tr>
    </table>
    </td>
    </tr>
</table>
</td>
</tr>
</center>
<BR>
<tr>
  <td height='20'> </td>
</tr><CENTER>
<tr>
<td align='center'>
    <input border='0' type='submit' name='addprocedimiento&index=<%= index
%>' value='Añadir procedimiento'>
</td>
```

```
</tr>

</CENTER>

</form>

<FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addtrata&index=<%=
index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >

<center>

 <table width="80%" border="1">

  <tr bgcolor="#8fbc8b"><th>Nombre
Procedimiento</th><th>CIE</th><th>Comentario</th></tr>

<center>

        <tr align=center>

           <td align='center'>

              <font ><%= request.getAttribute("Nombre0") %></font>

           </td>

           <td align='center'>

              <font ><%=request.getAttribute("Id0")%></font>

           </td>

           <td align='center'>

             <font ><%=request.getAttribute("Comentario0")%></font>

           </td>

        </tr>

<BR>

        <tr align=center>

           <td align='center'>

              <font ><%= request.getAttribute("Nombre1") %></font>

           </td>
```

```
            <td align='center'>
                <font ><%=request.getAttribute("Id1")%></font>
            </td>
            <td align='center'>
              <font ><%=request.getAttribute("Comentario1")%></font>
            </td>
         </tr>
<BR>
        <tr align=center>
            <td align='center'>
                <font ><%= request.getAttribute("Nombre2") %></font>
            </td>
            <td align='center'>
                <font ><%=request.getAttribute("Id2")%></font>
            </td>
            <td align='center'>
              <font ><%=request.getAttribute("Comentario2")%></font>
            </td>
         </tr>
<BR>
        <tr align=center>
            <td align='center'>
                <font ><%= request.getAttribute("Nombre3") %></font>
            </td>
            <td align='center'>
                <font ><%=request.getAttribute("Id3")%></font>
            </td>
```

```
            <td align='center'>
              <font ><%=request.getAttribute("Comentario3")%></font>
            </td>
          </tr>
</center>
</table>
<BR>
<tr>
  <td height='20'> </td>
</tr>
<tr>
  <td height='20'> </td>
</tr>
 <INPUT border="0" TYPE="SUBMIT" NAME="addtrata&index=<%= index %>"
VALUE="Continuar">
</FORM><BR>
<tr>
  <td height='20'> </td>
</tr>
<tr>
  <td height='20'> </td>
</tr>
<!-- Pie del documento -->
<%@ include file="footer.jsp" %>
```

addTratamiento.jsp

```
<%/* addTratamiento.jsp
 *
```

```
 * Proyecto Fin de Carrera 2002/2003

 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA

 * Y BASE DE DATOS XML NATIVAS

 *

 * @author  Irene Lavado Gomez

 */%>

<%/* En esta página jsp tendremos el formulario referente al tratamiento a seguir por el

 * paciente.  Cada vez que accedamos a esta página aparecerán los datos del paciente que

 * hayamos seleccionado, y modificando los campos que aparecen, podremos rellenar los campos,

 * para un nuevo paciente o modificarlos para un paciente con datos ya existentes

 */%>

<%@ include file="header2.jsp" %>

<jsp:include page="InformeAlta.EditTratamiento" />

<FORM  ACTION="/InformeAlta/servlet/InformeAlta.Task" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="ADDTRATAMIENTO">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

  <FONT SIZE="+2">

    <center>Añadir tratamiento</center>

  </FONT>

<tr>

<td>

<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360' >

   <tr>
```

```
    <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='360'>
         <tr>
         <td width='132' align='right' >
            <font >Fecha de alta: </font>
         </td>
         <td width='220' align='left'>
            <input type='text'name='FECHAALTA'
VALUE="<%=request.getAttribute("Fecha")%>" maxlength='30'
         </td>
         </tr>
      </table>
      </td>
      </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
```

```
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='360' >
   <tr>
    <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='360'>
         <tr>
         <td width='140' align='right' >
            <font >Tipo de alta: </font>
         </td>
         <td width='220' align='left'>
            <select                                          name='TIPOALTA'
VALUE="<%=request.getAttribute("Tipo")%>">
               <option
value='<%=request.getAttribute("Tipo")%>'><%=request.getAttribute("Tipo")%>
               <option value='Alta normal'>Alta normal
               <option value='Alta normal'>Alta voluntaria
               <option value='Alta irregular'>Alta irregular
               <option value='Exitus'>Exitus
            </select>
         </td>
         </tr>
         <tr>
         <td width='140' align='right' >
            <font >Lugar de alta: </font>
         </td>
```

```
        <td width='220' align='left' >

          <select                                    name='LUGARALTA'
VALUE="<%=request.getAttribute("Lugar")%>">

            <option
value='<%=request.getAttribute("Lugar")%>'><%=request.getAttribute("Lugar")
%>

            <option value='Exitus'>Exitus

            <option value='Residencia privada'>Residencia privada

            <option value='Institución'>Institución

            <option value='Lugar temporal'>Lugar temporal

            <option value='Traslado en el mismo hospital'>Traslado en el
mismo hospital

            <option value='Traslado a otro hospital público'>Traslado a otro
hospital público

            <option value='Otros'>Otros

          </select>

        </td>

        </tr>

      </table>

      </td>

      </tr>

   </table>

   </td>

   </tr>

</table>

</td>

</tr>

<tr>

<td > </td>
```

```
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
   <tr>
   <td>
     <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300'
>
      <tr>
        <td>
        <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
           <tr>
          <td width='300' align='left' >
             <font > Tratamiento médico:</font>
          </td>
          </tr>
          <tr>
          <td>
             <textarea rows=3 cols=30 name='TRATMEDICO'
><%=request.getAttribute("Trat")%></textarea>
          </td>
          </tr>
       </table>
        </td>
       </tr>
      </table>
      </td>
   </tr>
```

```
</table>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
   <tr>
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300' >
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
         <tr>
         <td width='300' align='left' >
            <font > Administración:</font>
         </td>
         </tr>
         <tr>
           <td>
            <textarea rows=3 cols=30 name='ADMINISTRACION'
><%=request.getAttribute("Adm")%></textarea>
         </td>
         </tr>
      </table>
```

```
          </td>
        </tr>
       </table>
       </td>
    </tr>
</table>
</td>
</tr>
<tr>
<td > </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
    <tr>
     <td>
    <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300' >
          <tr>
          <td>
       <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
           <tr>
           <td width='300' align='left' >
              <font > Observaciones finales:</font>
           </td>
           </tr>
           <tr>
           <td>
```

```
            <textarea rows=3 cols=30 name='OBSERVTRAT'
><%=request.getAttribute("ObservTrat")%></textarea>

          </td>

          </tr>

       </table>

       </td>

       </tr>

      </table>

   </td>

   </tr>

</table>

</td>

</tr>

<tr>

<td ><br><br></td>

</tr>

<tr><center>

<td align='center'>

   <input border='0' type='submit' name='addtratamiento&index=<%= index %>'
value='Anotar datos'>

</td>

</tr>

<tr>

<td > </td>

</tr>

<tr>

<td align='center'>

  <input border='0' type='reset' >
```

```
</td> </center>
</tr>
 </form>
 <tr>
  <td height='20'> </td>
</tr>
 <!-- Pie del documento -->
<%@ include file="footer.jsp" %>
```

editPersonales.jsp

```
<%/* editPersonales.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%/* En esta página jsp tendremos el formulario referente a los datos personales de un paciente
 * y tendremos la opción de modificarlos, en cuyo caso volveríamos al listado de pacientes,
 * o continuar visualizando, modificando o rellenando los datos referentes a ese paciente
 */%>
<%@ include file="header2.jsp" %>
<FONT SIZE="+2">
     <center>Datos Personales</center>
  </FONT>
```

```
<FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=editinforme&index=<%
= index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="editinforme">

   <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >

    <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

<tr>

<td>

<tr>

<td width='750'> </td>

</tr>

<tr>

<td>

   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360'>

 <tr><td>

    <table align='center' border='0' cellPadding='0' cellSpacing='0' width='360'>

      <tr>

          <td width='140' align='right'>

             <font >D.N.I.: </font>

          </td>

          <td width='220' align='left'>

             <input type='text' VALUE="<%= informe.getValorId() %>"
name='VALORID' maxlength='10'>

          </td>

          </tr>

          <tr>

          <td width='140' align='right'>

             <font>Nombre: </font>
```

```
        </td>
        <td width='220' align='left'>
          <input type='text' VALUE="<%= informe.getNombre() %>"
name='NOMBRE' maxlength='25' >
        </td>
        </tr>
        <tr>
        <td width='140' align='right'>
          <font>Primer apellido: </font>
        </td>
        <td width='220' align='left'>
          <input type='text' VALUE="<%= informe.getPrimerApellido() %>"
name="PRIMERAPELLIDO" maxlength='25'>
        </td>
        </tr>
        <tr>
        <td width='140' align='right'>
          <font>Segundo apellido: </font>
        </td>
        <td width='220' align='left'>
          <input type='text' VALUE="<%= informe.getSegundoApellido() %>"
name='SEGUNDOAPELLIDO' maxlength='25' >
        </td>
        </tr>
</table>
</td>
</tr>
</table>
</td>
```

```
</tr>

<tr>

<td width='750'> </td>

</tr>

<tr>

<td><br><br></td>

</tr>

<tr> <center>

<td align='center'>

   <INPUT  border="0"  TYPE="SUBMIT"  name='editinforme&index=<%=  index
%>' VALUE="Salvar datos">

</td>

<td> </td>

<td align='center'>

   <input border='0' type='reset'>

</td> </center>

</tr>

<tr>

  <td height='20'> </td>

</tr>

</form>

<center>

<FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=editdemo&index=<%=
index %>" METHOD="POST">

  <INPUT        TYPE="HIDDEN"        NAME="DOCKEY"        VALUE="<%=
informe.getDocKey() %>" >

 <INPUT  border="0"  TYPE="SUBMIT"  NAME="editdemo&index=<%=  index
%>" VALUE="Continuar">
```

```
</FORM>
</center>
<%@ include file="footer.jsp" %>
```

error.jsp

```
<%/* error.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%@ include file="header.jsp" %>
  <FONT SIZE="+5">
    <CENTER>
      <FONT COLOR="RED">Ha ocurrido un error!!!</FONT>
      <BR>
      <BR>
      <%
        String error = request.getParameter("error") ;
        if ( error != null ) {
          out.println( error );
        }
      %>
    </CENTER>
  </FONT>
<%@ include file="footer.jsp" %>
```

examenes.jsp

```
<%/* examenes.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%/* En esta página jsp tendremos la opción de elegir que tipo de examen queremos rellenar,
 * Exploración física, Hemograma, o Prueba bioquímica, o si deseamos continuar con el
 * formulario siguiente
 */%>
<%@ include file="header2.jsp" %>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300'>
   <tr>
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300'>
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='300'>
         <tr>
         <td width='300' align='left' >
```

```
            <font> Exploración física:</font>
          </td>
          </tr>
          <tr>
          <td align='center'>
            <br>
              <FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addexplor&index=<%=
index %>" METHOD="POST">
               <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >
               <INPUT border="0" TYPE="SUBMIT"
NAME="addexplor&index=<%= index %>" VALUE="Exploracion fisica">
              </FORM>
            <br><br>
          </td>
          </tr>
       </table>
       </td>
       </tr>
    </table>
    </td>
    </tr>
</table>
</td>
</tr>
<tr>
<td> </td>
</tr>
```

```
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300'>
   <tr>
   <td>
   <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300'>
      <tr>
      <td>
      <table align='center' border='0' cellPadding='1' cellSpacing='2' width='300'>
         <tr>
         <td width='300' align='left'>
            <font> Hemograma:</font>
         </td>
         </tr>
         <tr>
         <td align='center'>
            <br>
               <FORM ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addhemo&index=<%= index %>" METHOD="POST">
                <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >
                <INPUT border="0" TYPE="SUBMIT" NAME="addhemo&index=<%= index %>" VALUE="Hemograma">
               </FORM>
             <br><br>
         </td>
         </tr>
```

```
        </table>
        </td>
        </tr>
    </table>
    </td>
    </tr>
</table>
</td>
</tr>
<tr>
<td> </td>
</tr>
<tr>
<td>
<table align='center' border='0' cellPadding='0' cellSpacing='2' width='300' >
    <tr>
    <td>
    <table align='center' border='0' cellPadding='0' cellSpacing='0' width='300' >
        <tr>
        <td>
        <table align='center' border='0' cellPadding='1' cellSpacing='2'
width='300'>
            <tr>
            <td width='300' align='left' >
                <font > Prueba bioquímica:</font>
            </td>
            </tr>
            <tr>
```

```
        <td align='center'>
          <br>
            <FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addbioqui&index=<%=
index %>" METHOD="POST">
             <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >
             <INPUT border="0" TYPE="SUBMIT"
NAME="addbioqui&index=<%= index %>" VALUE="Pruebas Bioquimicas">
            </FORM>
          <br><br>
        </td>
        </tr>
      </table>
      </td>
      </tr>
   </table>
   </td>
   </tr>
</table>
</td>
</tr>
<tr>
<td>
   <br>
   <hr >
   <br>
</td>
</tr>
```

```
<tr>

<td align='center'>

<center>

<FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addjuic&index=<%=
index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="addjuic">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >

   <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= group.getIndex()
%>" >

  <input border='0' type='submit' name='addjuic&index=<%= index %>'
value='Continuar'>

</form>

</center>

</td>

</tr>

<tr>

<td height='20'> </td>

</tr>

<!-- Pie del documento -->

<%@ include file="footer.jsp" %>
```

footer.jsp

```
<%/* footer.jsp

 *

 * Proyecto Fin de Carrera 2002/2003

 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA

 * Y BASE DE DATOS XML NATIVAS

 *
```

```
 * @author  Irene Lavado Gomez
 */%>
                </td>
              </tr>
            </table>
          </td>
      </tr>
      </table>
    </td>
</tr>
</table>
<img align="center" src="../images/logo_junta.gif">
</body>
</html>
```

header.jsp

```
<%/* header.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%/* Esta es la cabecera de las páginas jsp mas generales, Pagina principal, Listado de
 * pacientes, Nuevo paciente y Buscar un paciente.  En ella creo un tabla con los posibles
 * enlaces a las opciones anteriores
```

```
 */%>

<%@ page errorPage="/error.jsp" %>

<html>

<head>

<title>Gestión del Informe de alta con tecnología Java y base de datos XML nativa</title>

</head>

<body text="#000000" bgcolor="#ffffff" background="" link="#191970" vlink="#551A8A" alink="FE0000" topmargin="10" leftmargin="10" marginwidth="10" marginheight="10">

<img src="../images/sas.gif">

<table border="0" cellspacing="2" cellpadding="0" width="95%" align="center" bgcolor="#FFFFFF">

  <tr>

    <td colspan="2">

    <table border="0" cellpadding="0" cellspacing="0" bgcolor="#8fbc8b" width="100%">

      <tr><td>

      <table border="0" cellspacing="1" cellpadding="2" width="100%" bgcolor="#8fbc8b">

        <tr>

          <td bgcolor="#8fbc8b" ALIGN="right">

            <font><CENTER>

              <A HREF="/InformeAlta">Pagina Principal</A> |

              <A HREF="/InformeAlta/servlet/InformeAlta.Task?action=listinf">Listado de informes</A> |

              <A HREF="/InformeAlta/servlet/InformeAlta.Task?action=addinf">Nuevo informe</A> |

              <A HREF="/InformeAlta/servlet/InformeAlta.Task?action=searchinf">Buscar un informe</A>
```

```
            </font></CENTER>
          </td>
        </tr>
      </table>
      </td></tr>
    </table>
    </td>
  </tr>
```

header2.jsp

```
<%/* header.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%@ page errorPage="/error.jsp" %>
<%@ page session="true" %>
<%@ page import="javax.servlet.http.HttpSession, java.io.* , java.util.*" %>
<html>
<head>
<title>Gestion del Informe de Alta con tecnología Java y base de datos XML nativa</title>
</head>
<jsp:useBean id="group" class="InformeAlta.Group"  scope="session" />
<jsp:useBean id="informe" class="InformeAlta.Informe" scope="session" />
```

```
<%

  int index = Integer.parseInt((String)request.getParameter("index"));

  informe = group.getInformeAt(index);

%>

<SCRIPT LANGUAGE="JavaScript">

</script>

<body text="#000000" bgcolor="#ffffff" background="" link="#191970" vlink="#551A8A" alink="FE0000" topmargin="10" leftmargin="10" marginwidth="10" marginheight="10">

<img src="../images/sas.gif">

<table border="0" cellspacing="2" cellpadding="0" width="95%" align="center" bgcolor="#FFFFFF">

  <tr>

    <td colspan="2">

    <table border="0" cellpadding="0" cellspacing="0" bgcolor="#5555FF" width="100%">

      <tr><td>

      <table border="0" cellspacing="1" cellpadding="2" width="100%" bgcolor="#5555FF">

        <tr>

          <td bgcolor="#8fbc8b" ALIGN="right">

            <font><CENTER>

              <A HREF="/InformeAlta">Pagina Principal</A> |

              <A HREF="/InformeAlta/servlet/InformeAlta.Task?action=listinf">Listado de informes</A> |

              <A HREF="/InformeAlta/servlet/InformeAlta.Task?action=searchinf&index=<%= index %>">Buscar un informe</A>  |

              <A HREF="/InformeAlta/servlet/InformeAlta.Task?action=deleteinforme&index=<%= index %>" onclick="return confirm('Estas seguro de que quieres borrar?')">Borrar un informe</A>
```

```
            </font></CENTER>
          </td>
        </tr>
      </table>
      </td></tr>
    </table>
    </td>
  </tr>
</table>

<table border="0" cellpadding="0" cellspacing="0" bgcolor="#FFFFFF" width="100%">

<tr >

<td><FORM ACTION="/InformeAlta/servlet/InformeAlta.Task?action=editdemo&index=<%= index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

<INPUT border="0" TYPE="SUBMIT" NAME="editdemo&index=<%= index %>" VALUE="Datos Demograficos">

</FORM></td>

<td>

<FORM ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addantece&index=<%= index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

<INPUT border="0" TYPE="SUBMIT" NAME="addantece&index=<%= index %>" VALUE="Antecedentes personales">

</FORM> </td>
```

```
<td><FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addcircunsdom&index=
<%= index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

<INPUT border="0" TYPE="SUBMIT" NAME="addcircunsdom&index=<%=
index %>" VALUE="Circunstancias al ingreso">

</FORM></td>

<td><FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=showxml&index=<%=
index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

<INPUT border="0" TYPE="SUBMIT" NAME="showxml&index=<%= index %>"
VALUE="Mostrar XML">

</FORM></td>

</tr>

</table>

<table border="0" cellpadding="0" cellspacing="0" bgcolor="#FFFFFF"
width="100%">

<tr>

<td>

<FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=exam&index=<%=
index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

 <INPUT border="0" TYPE="SUBMIT" NAME="exam&index=<%= index %>"
VALUE="Examenes realizados">

</FORM></td>
```

```
<td><FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addjuic&index=<%=
index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

<INPUT border="0" TYPE="SUBMIT"  NAME="addjuic&index=<%= index %>"
VALUE="Juicio Clinico">

</FORM></td>

<td><FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addproce&index=<%=
index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

 <INPUT border="0" TYPE="SUBMIT"  NAME="addproce&index=<%= index
%>" VALUE="Procedimiento Realizado">

</FORM></td>

<td><FORM
ACTION="/InformeAlta/servlet/InformeAlta.Task?action=addtrata&index=<%=
index %>" METHOD="POST">

  <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%=
informe.getDocKey() %>" >

  <INPUT TYPE="HIDDEN" NAME="INDEX" VALUE="<%= index %>" >

<INPUT border="0" TYPE="SUBMIT"  NAME="addtrata&index=<%= index %>"
VALUE="Tratamiento y Alta">

</FORM>

</td>

</tr>

</table>
```

header3.jsp

```
<%/* header3.jsp
```

```
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%/* indice para cuando estoy rellenando los datos de un paciente */%>
<%@ page errorPage="/error.jsp" %>
<%@ page session="true" %>
<%@ page import="javax.servlet.http.HttpSession, java.io.* , java.util.*" %>
<html>
<head>
<title>Gestion del Informe de Alta con tecnología Java y base de datos XML nativa</title>
</head>
<jsp:useBean id="group" class="InformeAlta.Group"  scope="session" />
<jsp:useBean id="informe" class="InformeAlta.Informe" scope="session" />
<%
  int index = Integer.parseInt((String)request.getParameter("index"));
  informe = group.getInformeAt(index);
%>
<img src="../images/sas.gif">
<body text="#000000" bgcolor="#ffffff" background="" link="#191970" vlink="#551A8A" alink="FE0000" topmargin="10" leftmargin="10" marginwidth="10" marginheight="10">
<table border="0" cellspacing="2" cellpadding="0" width="95%" align="center" bgcolor="#FFFFFF">
  <tr>
```

```
    <td colspan="2">
    <table border="0" cellpadding="0" cellspacing="0" bgcolor="#5555FF" width="100%">
      <tr><td>
      <table border="0" cellspacing="1" cellpadding="2" width="100%" bgcolor="#5555FF">
        <tr>
          <td bgcolor="#C8C8FF" ALIGN="right">
            <font>
            </font>
          </td>
        </tr>
      </table>
      </td></tr>
    </table>
    </td>
  </tr>
```

index.jsp

```
<%/* index.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
<%
/*
```

```
 */

%>

<%@ page import="javax.servlet.*, javax.servlet.http.*" %>

<%@ page errorPage="/error.jsp" %>

<html>

<head>

<title>Gestión de Informes de Alta con tecnología Java y base de datos XML nativas</title>

</head>

<body text="#000000" bgcolor="#ffffff" background="" link="#191970" vlink="#551A8A" alink="FE0000" topmargin="10" leftmargin="10" marginwidth="10" marginheight="10">

<table border="0" cellspacing="2" cellpadding="0" width="95%" align="center" bgcolor="#FFFFFF">

  <tr>

    <td colspan="2">

    <table border="0" cellpadding="0" cellspacing="0" bgcolor="#8fbc8b" width="100%">

      <tr><td>

      <table border="0" cellspacing="1" cellpadding="2" width="100%" bgcolor="#8fbc8b">

        <tr>

          <td bgcolor="#8fbc8b" ALIGN="right">

            <font><CENTER>

              <A HREF="/InformeAlta">Pagina Principal</A> |

              <A HREF="/InformeAlta/servlet/InformeAlta.Task?action=listinf">Listado de Pacientes</A> |

              <A HREF="/InformeAlta/servlet/InformeAlta.Task?action=addinf">Nuevo Paciente</A> |
```

```
                <A
HREF="/InformeAlta/servlet/InformeAlta.Task?action=searchinf">Buscar un
Paciente</A>

              </font></CENTER>

            </td>

          </tr>

        </table>

        </td></tr>

      </table>

      </td>

   </tr>

 </table>

</body>

</html>

<CENTER>

<B>

<H1>

IDENTIFICACION DEL USUARIO

</H1>

</B>

<IMG SRC="images/sas.gif"><p></CENTER>

<p><IMG ALIGN="left" SRC="images/logo_junta.gif"><IMG ALIGN="right"
SRC="images/ait.gif">
```

listInformes.jsp

```
<%/* listInformes.jsp

 *

 * Proyecto Fin de Carrera 2002/2003

 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
```

```
 * Y BASE DE DATOS XML NATIVAS

 *

 * @author  Irene Lavado Gomez

 */%>

<%/* En esta página jsp tendremos el listado de los informes que hay en la base de datos,

 * o un aviso de que la base de datos esta vacia.  Aqui tendremos que seleccionar el paciente

 * con el que deseamos continuar

 */%>

<%@ include file="header.jsp" %>

<%@ page session="true" %>

<%@ page import="javax.servlet.http.HttpSession, java.io.* , java.util.*" %>

<jsp:useBean id="group" class="InformeAlta.Group"  scope="session" />

<jsp:useBean id="informe" class="InformeAlta.Informe" scope="session" />

<BR>

<FORM NAME="editpersonales" ACTION="/InformeAlta/servlet/InformeAlta.Task?action=editpersonales" METHOD="POST">

 <INPUT TYPE="HIDDEN" NAME="DOCKEY" VALUE="<%= informe.getDocKey() %>" >

<%

   if (group.sizeOf() == 0) {

      %> No hay informes en la base de datos, esta vacia!!!! <%

   }

   else {

   %>

<center>

<FONT SIZE="+2">
```

```
  <CENTER>Listado de informes</CENTER>
</FONT>
</center>
<center>
 <table width="80%" border="1">
  <tr          bgcolor="#8fbc8b"><th>Selección</th><th>Nombre</th><th>Primer
Apellido</th><th>Segundo Apellido</th><th>Fecha de Alta</th></tr>
 <%
     group.reset();
     int i = 0;
     while ( group.hasMoreInformes() ) {
       informe = group.getNext();
       %>
<tr align=center>
          <td>
             <input type=radio name="index" value="<%= group.getIndex() %>" >
          </td>
          <td align="center">
            <%= informe.getNombre() %>
          </td>
          <td align="center">
            <%= informe.getPrimerApellido() %>
          </td>
          <td align="center">
             <%= informe.getSegundoApellido() %>
          </td>
          <td align="center">
            <%= informe.getFechaAlta() %>
```

```
          </td>
  <%
    }
  %>
 </table>
</center>
  <%
  }
%>
<P>
<center>
<FONT>
 <input border='0' type='submit' name="editpersonales&index=<%= group.getIndex() %>" value='Seleccionar Informe'>
</form>
<P>
</form>
<%@ include file="footer.jsp" %>
```

searchInforme.jsp

```
<%/* searchInforme.jsp
 *
 * Proyecto Fin de Carrera 2002/2003
 * GESTION DE INFORMES DE ALTA CON TECNOLOGIA JAVA
 * Y BASE DE DATOS XML NATIVAS
 *
 * @author  Irene Lavado Gomez
 */%>
```

```
<%
/*
  */
%>
<%@ include file="header.jsp" %>
<FONT SIZE="+2">
  <CENTER>
    Buscar un informe
  </CENTER>
</FONT>
  <TABLE BORDER="0">
    <FORM NAME="SEARCH" ACTION="/InformeAlta/servlet/InformeAlta.Task" METHOD="POST">
      <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="ATTRIBUTESEARCH">
      <TR>
        <TD COLSPAN="3">
          <FONT SIZE="+1">
            <B>
              Búsqueda por atributos
            </B>
          </FONT>
        </TD>
        <TD>
      </TR>
      <TR>
        <TD>
          <SELECT NAME="SEARCHTYPE">
```

```
            <OPTION SELECTED VALUE="ValorId">DNI
            <OPTION VALUE="Nombre">Nombre
            <OPTION VALUE="PrimerApellido">Primer Apellido
            <OPTION VALUE="SegundoApellido">Segundo Apellido
            <OPTION VALUE="FechaAdmision">Fecha Admision
          </SELECT>
        </TD>
        <TD>
          <INPUT TYPE="TEXT" NAME="SEARCHSTRING" SIZE="50">
        </TD>
        <TD>
          <INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="SEARCH">
        </TD>
      </TR>
      <TR>
        <TD>
           
        </TD>
      </TR>
    </FORM>
    <TR>
      <TD>
         
      </TD>
    </TR>
    <TR>
      <TD>
```

```
         
      </TD>
    </TR>
    <TR>
      <TD>
         
      </TD>
    </TR>
    <TR>
      <TD>
         
      </TD>
    </TR>
    <FORM NAME="XPATHSEARCH"
ACTION="/InformeAlta/servlet/InformeAlta.Task" METHOD="POST">
      <INPUT TYPE="HIDDEN" NAME="ACTION" VALUE="XPATHSEARCH">
      <TR>
        <TD COLSPAN="3">
          <FONT SIZE="+1">
            <B>
              Búsqueda Xpath especificando camino completo
            </B>
          </FONT>
        </TD>
      </TR>
      <TR>
        <TD>
          Búsqueda Xpath
```

```
          </TD>
          <TD>
            <INPUT TYPE="TEXT" NAME="SEARCHSTRING" SIZE="50">
          </TD>
          <TD>
            <INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="SEARCH">
          </TD>
        </TR>
        <TR>
          <TD>
             
          </TD>
        </TR>
      </FORM>
    </TABLE>
<!-- Pie del documento -->
<%@ include file="footer.jsp" %>
```

7 - APÉNDICE A

BIBLIOGRAFÍA

Libros

- **XML**. Natanya Pitts. Anaya Multimedia - Temas Profesionales
- **XML al descubierto**. Michael Morrison. Prentice Hall
- **JAVA y XML**. Brett McLaughlin. Anaya Multimedia

Direcciones de internet

- http://www.rae.es: Diccionario de la Real Academia Española de la Lengua. Portal en línea.
- http://www.w3.org/MarkUp: Especificación oficial de W3C para HTML.
- http://www.programacion.com/html/xml: Tutorial de XML en línea.
- http://www.programacion.com/tutorial.schemaydtd.html: Tutorial de DTD y XML Schema en línea, así como de otras tecnologías afines.
- http://www.w3.org/TR/xmlschema-0: Recomendación oficial de W3C para XML Schema.
- http://www.osmosislatina.com/xml/domsax.htm: Tutorial de DOM & SAX en línea.
- http://www.simba.vsf.es/FrameDoc.htm: Tutorial de DOM & SAX en línea.
- http://www.apache.net: Página principal del servidor *web* Apache. Proporciona las fuentes y diversas facilidades para el programador.
- http://www.sidar.org/recur/desdi/traduc/es/xml/xpath.html: Recomendación del W3C sobre Lenguaje de Caminos XPath.
- http://bob.marlboro.edu/~msie/2003/db2/s5.html: Ejemplos de DOM y Xpath.
- http://www.programacion.com/tutorial/javaxml/: Comparación de las tecnologías Java para XML.

- http://www.programacion.com/java/tutorial/jspyxml/: Desarrollo de Aplicaciones Web con JSP y XML.
- http://programacion.com/tutorial/apuntesxml/: Apuntes de XML.
- http://javaalmanac.com/egs/index.html: Ejemplos de Java.
- http://www.xmldb.org/xapi/api/org/xmldb/api/: APIs de dbXML.
- http://java.sun.com/j2se/1.4.2/docs/guide/plugin/dom/org/w3c/dom/package-use.html#org.w3c.dom: org.w3c.dom
- http://www.programacion.com/bbdd/articulo/xmlapisdata/: APIs XML para bases de datos.

8 - APÉNDICE B

INSTALACIÓN DE XINDICE

Antes de poder poner a funcionar o usar Xindice en windows es necesario obtener e instalar una versión 1.3 o superior de Java SDK.

El código puede descargarse de forma gratuita de www.apache.org y una vez descargado el proceso a seguir sería:

1) Descomprimir la distribución Xindice en el directorio elegido, supongamos c:\xindice

2) Asegurarse de que el directorio bin del JDK este en el PATH.

3) Añadir el directorio bin de Xindice al PATH. Estará en c:\xindice\bin.

4) Crear las variables JAVA_HOME y XINDICE_HOME, que apuntan a los directorios donde fueron instalados JDK y Xindice, respectivamente.

5) Añadir el fichero Xindice.jar al CLASSPATH (c:\xindice\java\lib\xindice.jar).

6) Para poner en funcionamiento el servidor, colocarse en la base del árbol donde Xindice fue instalado, c:\xindice, y

```
startup
```

7) Si todo va bien, tendremos una copia de Xindice corriendo.

8) Para crear una colección y probar la instalación corriendo sobre ella ejemplos.

```
xindiceadmin ac –c /db –n addressbook
```

Si funciona bien debería verse:

```
Created: /db/addressbook
```

Esto crea una colección llamada “addressbook” en la base de datos.

9) El servidor y la documentación pueden ser accedidos usando el browser y apuntando a:

```
http://localhost:4080/
```

Para añadir una nueva versión de Tomcat a Sun ONE el proceso es:

1) En el Explores del Editing, pinchar con el botón derecho sobre el nodo Tomcat 4.0, bajo el nodo Installed Servers y elegir Add Tomcat 4.0 Installation del menú.

2) En el campo Home dir de la pantalla de diálogo de Add Tomcat 4.0 Installation, poner la dirección del Tomcat 4.0 que se desea añadir. Tener en cuenta que el sistema verificará que se le ha especificado un directorio home Tomcat 4.0 válido.

3) En el campo Base dir, poner el tipo o browse de la base del directorio del servidor, si especificar un directorio Base es necesario.

4) Elegir un modo de integración IDE, seleccionando el botón apropiado. El valor por defecto es Full. En modo Minimum, el IDE no modifica ningún fichero en la instalación Tomcat. En este modo el IDE no podrá incluir monitorización http, compilación JSP, y debugging JSP.

9 - APÉNDICE C

W3C

El W3C fue fundado en octubre de 1994. Su cometido es desarrollar protocolos de uso común a través de un "consorcio internacional de la industria". http://www.w3.org/

En el W3C, se puede encontrar:

- Un repositorio de información acerca del WWW para desarrolladores y usuarios.
- Código de referencia para la implementación de los distintos estándares.
- Prototipos diversos y aplicaciones de ejemplo para demostrar el uso de nuevas tecnologías fomentadas por el W3C.

El W3C es una colección de grupos interesados que se reúnen para discutir acerca de nuevas tecnologías en Internet. Su proceso es bastante considerable y profundo, y tiene como resultado una decisión consensuada (aunque existen historias en el web de decisiones tomadas a la fuerza dentro de los grupos de trabajo); de cualquier forma el resultado final tiene un gran impulso.

¿Cómo trabaja el W3C?

Primero alguien (un grupo, una compañía, una persona) eleva una nota acerca de una tecnología o un asunto técnico al W3C. Después de un montón de debate y consultas, se crea una propuesta de borrador. Con suerte y mucho trabajo, se puede llegar a una recomendación. En este momento, la tecnología se considera lista para su adopción en la industria.

Nota

Una nota es simplemente una idea, no existe compromiso para seguir adelante con una nota.

Borrador de trabajo

Este es un trabajo en progreso, un grupo dentro del W3C está activamente trabajando en él.

Propuesta de Recomendación

Algún consenso ha sido logrado dentro del grupo de trabajo. Se ha elevado para revisión.

Recomendación

La idea que comenzó como una nota ahora se ha vuelto 'oficial'. Según el W3C, esta tecnología está lista para su uso.

Qué no es el W3C

El W3C no hace las veces de policía. No impone los acuerdos hechos o las tecnologías diseñadas. Cada una de las partes involucradas puede (y a menudo lo hacen) "extender" o mejorar los diseños.

Más información

Para ver algunas de las actividades y proyectos dentro del W3C, dirijirse a http://www.w3.org/Consortium/Activities

.

10-APÉNDICE D

Proyecto TASS

Entre los proyectos de tarjeta inteligente más importantes en los que ha tomado parte activa la FNMT-RCM destaca el proyecto TASS. Mediante este proyecto, el Ministerio de Trabajo y Seguridad Social decidió crear un documento que facilitara las relaciones entre los ciudadanos y la Administración Pública. Este documento, en formato de tarjeta inteligente, ayudará al Ministerio a reducir los tiempos de respuesta en sus actividades, así como los costes derivados del procesamiento de información, a la vez que incrementará la variedad de servicios ofrecidos a los ciudadanos.

Los servicios que el Gobierno español ofrece a sus ciudadanos están clasificados en el orden económico y sanitario (beneficios de desempleo, pensiones...). El objetivo del proyecto es el de suministrar una tarjeta a cada uno de los ciudadanos, permitiendo al usuario identificarse ante diferentes proveedores de servicios de una forma eficiente, fácil y muy cómoda.

La tarjeta también permitirá al usuario la posibilidad de obtener información acerca de las pensiones, inicio y finalización de enfermedades, desempleo..., así como obtener diferentes certificados. Estas operaciones requieren una identificación securizada del ciudadano, que sólo es posible utilizando un sistema de reconocimiento biométrico.

La responsabilidad de la producción, personalización y distribución de estas tarjetas fue conferida a la FNMT-RCM, ejerciendo también un papel decisivo en la etapa de diseño, planificación del proyecto e implantación operativa.

Para comprobar las distintas opciones técnicas y la eficacia de su propósito fue lanzado el proyecto piloto durante 1995 en la provincia de Córdoba. Dicho proyecto piloto significó la fabricación, por parte de la FNMT-RCM, de 500.000 tarjetas inteligentes de 3KB.

Tras el éxito alcanzado en esta fase piloto, la FNMT-RCM continúa con la emisión global con el fin de alcanzar los objetivos del proyecto nacional. Hasta ahora la FNMT-RCM ha fabricado ya alrededor de 7.000.000 tarjetas.

Proyecto diraya

El objetivo del proyecto Diaria es integrar toda la información sanitaria de cada ciudadano de forma que:

1) Sea disponible y pertinente para la atención sanitaria en el lugar y momento en que el usuario lo precise.

2) Facilite el trabajo de los profesionales.

3) Permita una explotación homogénea y útil para todos los niveles, funciones y organizaciones del Sistema Sanitario Público (planificación, gestión, evaluación o investigación).

En definitiva, la finalidad de Diraya es obtener e implantar, mediante tecnologías informáticas, una Historia de Salud única para cada ciudadano en el ámbito de Andalucía. Inicialmente su ámbito es la Atención Primaria ya que, en gran medida, viene a sustituir las herramientas aportadas por el proyecto TASS, aprovechando la experiencia obtenida con el mismo. Se trata de dar un paso adelante dotando de integridad a los sistemas.

Diraya es un proyecto integrado por diversos servicios, como el empleo de la tarjeta sanitaria individual como sistema de identificación y acceso al sistema sanitario, la creación de la Base de Datos de Usuarios (BDU) para todos los ciudadanos andaluces, la puesta en marcha de una Historia digital de salud única o la oficina virtual Inters@s, que facilita que los ciudadanos realicen distintas gestiones sanitarias a través de internet. La BDU se terminó en octubre de 2001 y contiene información básica de todos los andaluces, accesible en la red con el número de la tarjeta individual sanitaria. El programa Receta XXI también está integrado en el proyecto Diraya y afectará a los farmacéuticos de forma especial.

1. BASE DE DATOS DE USUARIOS (BDU)

La base de datos de usuarios es la columna vertebral del proyecto Diraya. Tiene las siguientes características:

- Está gestionada por una aplicación web lo que significa que no precisa de instalación de software local.

- Se nutre de la base de datos de la Seguridad Social, con la que se realiza un cruce mensual. De esta forma se asegura la actualización de los datos de aseguramiento que, por Ley, dependen de la situación en Seguridad Social (p.ej. el derecho a la prestación farmacéutica). También se realizan cruces con el Índice Nacional de Defunciones y Bases de datos de Mutualidades (MUFACE, MUJEGU, ISFAS). Tras estos cruces se realizan procesos de detección de incongruencias y duplicados.

- Identifica de manera unívoca a los ciudadanos. Permite conocer la situación de aseguramiento de los mismos y su adscripción al médico de atención primaria.

- Localiza la información sanitaria de los ciudadanos y actualiza los datos administrativos de las bases de datos locales de los centros.

- Incorpora la gestión de los desplazamientos de los ciudadanos de su domicilio habitual.

- Permite la elección individual de médico, rompiendo con la dependencia actual de los beneficiarios respecto de su titular. Asimismo es la herramienta básica para la gestión del nuevo modelo retributivo del personal sanitario de EBAP, que entró en vigor por Decreto 260/2001 de 27 de noviembre.

- Facilita la gestión de los cupos médicos permitiendo su apertura o cierre de manera parcial o total, y simplificando la redistribución de cupos.

- Incluye un directorio de centros asistenciales estructurado jerárquicamente.

Durante el año 2001, el proyecto pasó la fase de pilotaje, se realizaron los cursos de formación de operadores, y finalmente entró en producción en todos los Distritos de atención primaria el día 22 de octubre.

Se ha constituido en eje de todos los sistemas de información centrados en el usuario. De esta forma son ya varios los sistemas que interactúan o interactuarán con BDU para obtener la identificación de los usuarios o para obtener información de centros sanitarios: Historia Digital de Salud, Vigilancia Epidemiológica, Sistemas de Admisión Hospitalarios, Portal de Salud, Catálogo de centros, Gestión de la Demanda, Centro de Información y Servicios al Ciudadano, Receta XXI.

2. HISTORIA DIGITAL DE SALUD DEL CIUDADANO

Durante el año 2001 se acometió el desarrollo de la aplicación de Historia Digital de Salud sustituiría en los centros de atención primaria a la que se desarrolló en el marco del proyecto TASS. Han participado en el mismo unos 200 profesionales de todas las categorías profesionales y niveles de atención implicados. Constituirá el embrión de la historia única para todo el Sistema Sanitario Público de Andalucía. Su implantación comenzó en 2002.

Las características son:

• Única por paciente. Accesible desde cualquier centro sanitario.

• Vinculada al uso de la tarjeta sanitaria, que se convierte en la llave necesaria para salvaguardar la confidencialidad.

• Administración mixta: local y central, lo que permite la homogeneidad de las tablas auxiliares.

• Integrada con BDU para asegurar la consistencia de los datos administrativos de los ciudadanos.

• Preparada para la comunicación entre atención primaria y hospitales (analíticas, pruebas, informes…) facilitando la continuidad asistencial (atención por procesos).

• Basada en la posibilidad de agrupación de pacientes para la medición del producto sanitario.

• Útil para todos los profesionales de atención primaria.

• Configuración personalizada de escritorios.

3. MÓDULO DE TRATAMIENTO DE LA INFORMACIÓN (MTI)

Es un Sistema Integrado de Gestión de Información basado en sistemas Data Warehouse y tecnologías OLAP. Sus características son:

• Extrae la información relevante de los sistemas de gestión a un entorno más óptimo orientado a la consulta y al análisis, y separado de los sistemas operacionales. Esto mejora los tiempos de respuesta.

• Integra la información proveniente de los distintos sistemas o módulos para permitir la consulta combinada de ellos.

• Aisla al operador final de los detalles técnicos con los que se representa la información.

• Permite el acceso a un gran número de operadores (tecnología web).

• Se obtienen informes previamente definidos y se pueden realizar consultas complejas a medida. Crea cuadros de mando.

• Gestiona el sistema dentro de las Tecnologías de Relación, orientando la información hacia el usuario y podrá, en un futuro, proporcionar información al mismo a través del Centro de Información y Servicios al Ciudadano.

Inicialmente, las fuentes de información son BDU e Historia de Salud, aunque será extensible a todas las aplicaciones que tengan como eje principal a los usuarios del Sistema Sanitario Público Andaluz. Durante el año 2001 se ha completado el desarrollo de la 1ª fase del MTI, que recoge la información de BDU, quedando lista para su entrada en producción. Asimismo, se han desarrollado los requisitos de la 2ª fase que contempla los datos de la Historia de Salud.

Receta XXI

Receta XXI permitirá a los pacientes adquirir los fármacos en su farmacia sin necesidad de receta física y mediante su tarjeta individual sanitaria. Los meses de julio y agosto los dedicará el SAS a formar a los médicos en la nueva aplicación informática, indica Aguilar.

La formación de los farmacéuticos que participarán en la experiencia corre a cargo de los colegios profesionales en colaboración con la gestora del programa informático. Todavía no ha empezado la formación de los farmacéuticos.

Las primeras pruebas con pacientes y recetas reales del programa Receta XXI del Servicio Andaluz de Salud (SAS), la primera experiencia piloto,

conectará el Centro de Salud de Torreblanca, un barrio de Sevilla, con las ocho farmacias que corresponden a su zona.

Pacientes virtuales

También se realizarán pruebas con pacientes y recetas virtuales en el centro de salud de Torreblanca. Cuando llegue la hora de la verdad y de aplicar la nueva herramienta con recetas reales, la experiencia se hará de forma progresiva. Se seleccionará a "diez pacientes para que usen la receta electrónica". Después se aumentará paulatinamente el número de usuarios en este centro de salud sevillano. Al mismo tiempo, "se extenderá el pilotaje al resto de las provincias, seleccionando un centro de salud y las farmacias que le corresponden".

Para implantar Receta XXI, es necesario que en los centros sanitarios prescriptores esté instalada la Historia de Salud Digital del Ciudadano, es decir, la historia clínica electrónica. La atención primaria de Andalucía ya está informatizada y conectada en 1.300 redes locales, por lo que se prevé que a finales de este año Receta XXI esté en "gran parte de los centros de salud", ha afirmado la responsable de Farmacia del SAS.

Para la aplicación en atención especializada queda un poco más de tiempo, ya que ahora se están desarrollando las primeras experiencias de la historia única informatizada entre centros de salud y hospitales. Se espera que "a lo largo de 2004" se implantará en AE.

Si se cumplen los plazos, Andalucía se convertirá en la primera autonomía en adoptar este sistema para la dispensación, que "ahorra tiempo a médicos, farmacéuticos y usuarios".