

Digitalització Imatge Mèdica No Radiològica

JAVIER PÉREZ MARTÍN
PRIMAVERA 2010

Digitalització Imatge Mèdica no Radiològica



Enginyeria Tècnica en Informàtica de Gestió

Escola Universitària
Politécnica de Mataró





Escola Universitària
Politécnica de Mataró

Enginyeria Tècnica en Informàtica de Gestió

DIGITALITZACIÓ D'IMATGE NO RADIOLÒGICA

**JAVIER PÉREZ MARTÍN
JOSEP ROURE ALCOBÉ**

PRIMAVERA 2010

RESUM

Digitalització de la imatge mèdica no radiològica

El projecte de la digitalització de la imatge mèdica va començar fa uns anys amb la imatge mèdica radiològica però el projecte de la digitalització encara no acabat ja que encara falta per digitalitzar tota la part de la imatge no radiològica que estaria formada per el electrocardiogrames, les espirometries, els pulsòmetres, els audiòmetres, les proves d'esforç...tota aquesta informació actualment no es guarda al sistema d'informació dels hospitals i en els casos que si, només es guarda en format pdf de tal manera que aquesta informació posteriorment no es possible manipular-la.

El projecte consisteix en la digitalització de les proves d'esforç que són una eina de detecció per avaluar l'efecte de l'exercici sobre el cor. Aquest examen ofereix una visió general de la salut del cor.

En el projecte s'utilitzaria l'aparell de General Electrics (GE) ja que és l'aparell més comú en els centres hospitalaris, l'aparell de GE extreu un fitxer en format xml que seria el que utilitzaríem per transformar-lo a format DICOM (Digital Imaging and Communication in Medicine), que és l'estàndard reconegut mundialment per l'intercanvi d'imatges mèdiques. Aquest format està pensat per al maneig, emmagatzematge, impressió i transmissió d'imatges mèdiques que és el tipus de fitxer que permet emmagatzemar el PACS (acrònim de Picture Archiving and Communication System; és a dir, és un sistema d'arxiu i distribució d'imatge mèdica digital).

Aparell de GE → Fitxer xml → **CONVERTIDOR** → Fitxer DICOM → PACS

L'aplicació està desenvolupada en JAVA

RESUMEN

Digitalización de la imagen médica no radiológica

El proyecto de la digitalización de la imagen médica radiológica, empezó hace unos años con la digitalización de la imagen médica radiológica, pero el proyecto de la digitalización todavía no ha acabado ya que falta por digitalizar toda la parte que corresponde a la imagen médica no radiológica, que estaría formada por: los electrocardiogramas, los pulsímetros, los audímetros, las espirómetros, las pruebas de esfuerzo... toda esta información, actualmente no se guarda en el sistema de información de los hospitales y en los casos que sí, únicamente se guarda en formato pdf, lo que hace que no sea posible manipularla.

El proyecto consiste en la digitalización de la pruebas de esfuerzo, que son una herramienta de detección para evaluar el efecto del ejercicio en el corazón. Este examen ofrece una visión general de la salud del corazón.

En el proyecto se utilizaría la máquina de General Electrics (GE), ya que es la más común en los centros hospitalarios, dicha máquina exporta un fichero en formato xml que sería el que utilizaríamos para transformarlo a formato Dicom (Digital Imaging and Communication in Medicine), que es el tipo de fichero que permite almacenar el PACS (acrónimo de Picture Archiving and Communication System; es decir, es un sistema de archivo y distribución de imagen médica digital).

Máquina de GE → Fichero xml → **CONVERTIDOR** → Fichero DICOM → PACS

La aplicación está desarrollada en Java.

Abstract

Digitalization of non-radiological medical imaging

The digitalization of radiological medical imaging was a project initiated a few years ago with the digitalization of radiological medical imaging. Nevertheless, the project is not over yet, since information on non-radiological imaging –electrocardiograms, heart rate monitors, audiometers, spirometers, stress tests...– is not saved in the hospitals databases and, in the cases when it is stored, it is only kept as PDF files, so it cannot be manipulated.

The project includes the digitalization of stress tests, which is a screening tool to assess the effect of exercise over the heart. This test provides an overview of heart health.

The project will use General Electrics machine (GE), given it is the one more frequently found in hospitals; it will export a XML file, which will later be converted to DICOM (Digital Imaging and Communication in Medicine), the type of file used to store the PACS (Picture Archiving and Communication System, that is, a filing and distribution of digital medical imaging application).

GE machine -> XML file -> **CONVERTER** -> DICOM file -> PACS

This application is developed in Java.

INDEX

Motivació del projecte.	I - IV
Objectius del projecte.	V - VI
Especificacions tècniques	
Funcionalitats a desenvolupar.	VII
Disseny de l'aplicació.	VIII
Estructura del convertor.	IX - X
Estructura de carpetes.	XI
Fil d'execució d'un arxiu.	XII - XIII
Diagrames de Classes	
Diagrama de classes Demoni.	XV
Diagrama de classes Conversion Process	XVI
Diagrama de classes ECG to DCM : part 1	XVII
Diagrama de classes ECG to DCM : part 2	XVIII
Digrama de classes Validació	XIX
Diagrama de classes Estructura Dicom	XX
Diagrama de classes Crear Dicom	XXI
Diagrama de classes Logs	XXII
Diagrama de classes Complet	XXIII
Diagrames de Seqüència	
Diagrama de seqüència Dimoni	XXV
Diagrama de seqüència Conversion Process	XXVI

Diagrama de seqüència Conversion Validació Xml	XXVII
Diagrama de seqüència Conversion Ecg To Dcm	XXVIII
Diagrama de seqüència Estructura Dicom	XXIX
Manual d'usuari	
Instal·lació del Conversor	XXXI
Requeriments de software	XXXII
Integració de Fabricants	
Fabricants integrats en l'aplicació	XXXIII
Incorporació d'un nou fabricant	XXXIII
Standard Dicom	
Explicació	XXXV
Dicom versió 3.0	XXXVI - XXXVII
Estructura dels fitxers Dicom	XXXVIII - XLV
Codificació d'un fitxer Dicom	XLVI - XLVIII
Llibreries Dicom	
Llibreries Conegudes	XLIX - L
Dcm4Chee	
Explicació	LI
Viabilitat implementació conversor mitjançant XSLT	LIII
Model de dades Dicom.	LV - LVIII
Equivalència Variables Xml de GE amb Tags Dicom.	LIX - LXIV
Conclusions	LXV
Bibliografia	LXVII

MOTIVACIÓ DEL PROJECTE

Aquest projecte neix amb la idea de contribuir a la digitalització de la imatge mèdica no radiològica a Catalunya . Una de les finalitats d'aquest projecte consisteix en millorar la interoperabilitat dins del mateix hospital, i la interoperabilitat d'un hospital amb altres.

En aquesta primera part del projecte donaré una visió general de la situació en que es troben la gran majoria dels hospitals de Catalunya en els temes relacionats amb la integració d'imatge mèdica no radiològica en els seus sistemes d'informació .

Considerarem imatge mèdica no radiològica totes les proves mèdiques en les que els fitxes que s'obté quan finalitza una prova mèdica en comptes de ser una imatge estan compostades per diferents tipus de gràfics, com poden ser les ones d'una prova d'esforç.

Les màquines que extreuen imatge mèdica no radiològica serien: Electrocardiogrames, Monitors de UCI, Espiròmetres, Pulsímetres, Audiòmetres Ventiladors Mecànics...

Actualment en el hospitals hi ha una gran quantitat de màquines per l'adquisició d'imatges i ones, la majoria de la imatge mèdica radiològica està solucionada (els fabricants importants segueixen els estàndards) però en la imatge mèdica no radiològica la situació és una mica més complicada, també tenim molts fabricants amb la dificultat que cadascun d'ells implementa en els seus fitxers en un format propietari, el que dificulta la comunicació amb els sistemes dels hospitals.

Per poder aconseguir aquesta fita hem tingut que escollir un estàndard per codificar el tipus de prova que expliquem en apartats posteriors, s'han barallat varies opcions per triar l'estàndard i finalment s'ha optat per l'estàndard Dicom, que és el que actualment s'està usant per emmagatzemar imatge mèdica radiològica.

Hem centrat el projecte en la digitalització de les proves; una prova d'esforç consisteix en una valoració de la resposta d'un organisme durant l'exercici, permeten veure com funciona el cor durant l'exercici.

Quan es realitza un prova d'esforç, als hospitals hi han dos solucions per guardar la informació proporcionada per l'aparell:

- 1- Una vegada la prova ha finalitzat el propi aparell fa una impressió dels resultats obtinguts i aquest document s'arxiva amb la resta d'historial de la persona.

- 2- Hi ha hospitals que acaben comprant sistemes del propi fabricant de l'electrocardiograma per guardar les proves:

Problemes:

- i. Aquest sistema només acceptarà fitxers seus, si l'hospital té diferents màquines d'electrocardiografia (diferents fabricants) la resta de fitxers dels altres fabricant no es podrà guardar al sistema.

- ii. Els fitxers guardats sinó segueixen l'estàndard DICOM no podran ser visualitzats amb visors genèrics, només es podran veure amb el visor que proporcioni el fabricant.

Flux a l'hora de realitzar una prova d'esforç a un hospital, quan un metge no es troba a l'hospital:

1-La infermera realitza la prova d'esforç

2- L'aparell extreu un paper amb les ones i diferents valoracions proporcionades per la màquina , un fitxer que és enviat a una base de dades (el fitxer està en format propietari) o es produeixen les dues situacions.

3-L'infermera envia via fax la informació del paper al metge responsable del pacient

(Dos problemes principals: una gran pèrdua de qualitat en la definició de les ones, dificultats per mesurar...)

4-El metge diagnostica amb el paper que li ha arribat via fax.

Flux a l'hora de realitzar una prova d'esforç a un hospital (el metge està a l'hospital i la prova d'esforç es guarda en paper):

1-La infermera realitza la prova d'esforç

2-L'aparell extreu un paper amb les ones i diferents valoracions proporcionades per la màquina , un fitxer que és enviat a una base de dades (el fitxer està en format propietari) o es produeixen les dues situacions.

3-El paper es guarda amb l'historial del pacient

(dificultats per mesurar, no es poden fer modificacions amb la prova)

4-El metge diagnostica amb el paper que posteriorment li porta la infermera.

Flux a l'hora de realitzar una prova d'esforç a un hospital (el metge està a l'hospital i la prova d'esforç es guarda en format propietari):

1-La infermera realitza la prova d'esforç

2-L'aparell extreu un paper amb les ones i diferents valoracions proporcionades per la màquina , un fitxer que és enviat a una base de dades (el fitxer està en format propietari) o es produeixen les dues situacions.

4-El metge té que anar a buscar l'ordinador que conté el visor específic per poder visualitzar el fitxer.

Donada aquesta situació hem vist la necessitat de estandarditzar la situació, és a dir, convertir tots els arxius que extreuen aquestes màquines a un format estàndard (DICOM) amb la finalitat d'aconseguir que les proves puguin ser emmagatzemades al PACS i posteriorment visualitzades des de qualsevol part de l'hospital amb un visor genèric (anomenem visor genèric a aquells visors que poden visualitzar un format estàndard).

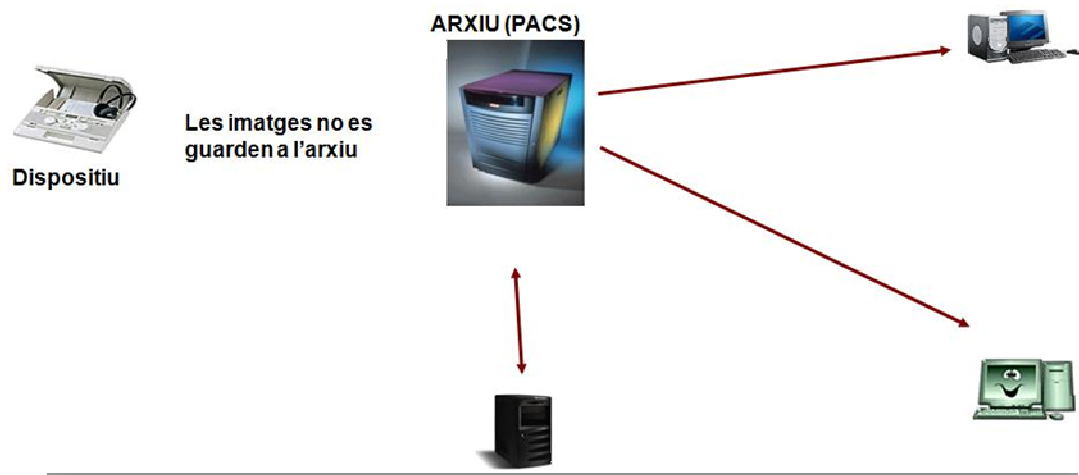
El PACS és una combinació de hardware i software dedicat a permetre guardar, gestionar i distribuir imatges. El format universal per a l'emmagatzematge d'imatges al PACS i la transferència és DICOM (Digital Imaging and Communication in Medicine).

D'aquesta manera millorarem la interoperabilitat dintre d'un mateix centre i també amb altres centres, la informació podrà ser visualitzada des de qualsevol ordinador (dintre hospital o fora) ja que els visors genèrics com RaimJava (visor que actualment s'està utilitzant per altres tipus de proves) no necessàriament tenen que estar instal·lats a l'ordinador ja que està creat com un servei web.

OBJECTIUS DEL PROJECTE

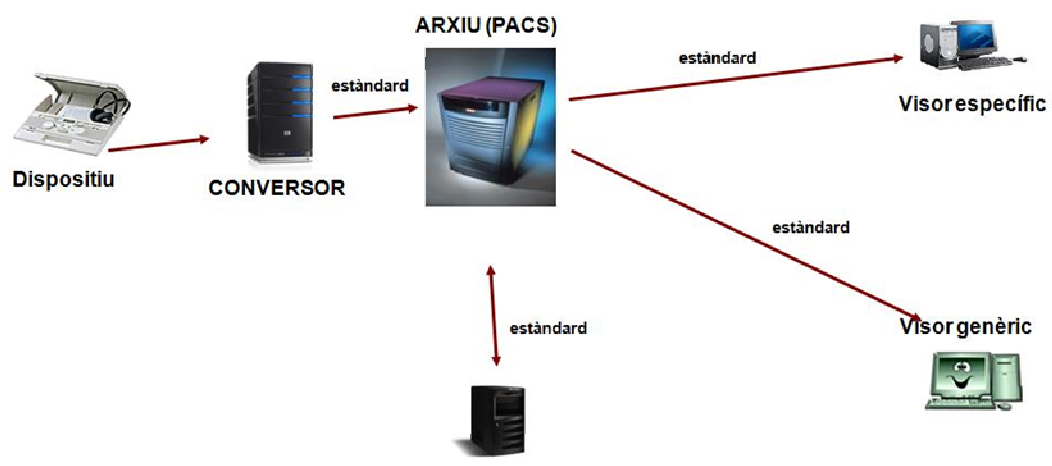
L'objectiu del projecte consisteix en la creació d'una aplicació per passar un fitxer que conté informació en format propietari a un format estàndard, l'aplicació s'ha construït amb la finalitat que altres fabricant puguin acoblar-se d'una manera molt simple, l'aplicació actualment només funcionaria per un fabricant (General Electrics). El convertor agafaria un fitxer (xml) d'una carpeta, el processaria i el deixaria en format DICOM en una altre carpeta, llest per ser enviat cap al PACS.

Si representem gràficament la situació en que es trobaria un hospital actualment seria com la que representem en el gràfic de sota:



L'hospital te moltes màquines (en el nostre cas per a fer proves d'esforç) però la informació no s'emmagatzema al PACS. Els fitxers resultants únicament es podrien emmagatzemar en paper, o en algun sistema tipus propietari.

Aquesta seria la situació a la que es pretén arribar amb el convertidor:



D'aquesta manera els fitxers s'emmagatzemarien al PACS (fitxers que segueixen l'estandard DICOM) i una vegada emmagatzemats es podrien visualitzar tants des d'un visor específic com des d'un visor genèric.

ESPECIFICACIONS TÈCNIQUES

Funcionalitats a desenvolupar:

L'objectiu és desenvolupar una aplicació que rebí un fitxer en format xml (l'aplicació estarà escoltant una carpeta) i transformi aquest fitxer en un fitxer amb format dcm.

L'aplicació es dissenyarà com un "daemon", és a dir, s'estarà contínuament executant a l'espera de rebre nous fitxers xml a una carpeta determinada per transformar-los en format DICOM.

Aquesta aplicació s'instal·larà com un servei Windows, d'aquesta manera no farà falta que l'engeguem cada cop que es reiniciï el servidor o es tanqui la sessió d'un usuari.

L'aplicació serà totalment invisible cara a l'usuari, només s'haurà de programar l'exportació de la màquina cap a la carpeta que la nostre aplicació estarà escoltant. Posteriorment l'aplicació realitzarà la conversió i un altre programa s'encarregarà d'enviar el fitxer cap al PACS.

Quan l'aplicació rebí un arxiu que estigui en un format incorrecte, aquest serà automàticament desplaçat cap a una altre carpeta (xmlErroneos), d'aquesta manera no perdrem l'arxiu i podrem veure quin ha estat el motiu de l'error.

L'aplicació ha de ser capaç de tenir una gran tolerància a fallades sense que es produeixi cap pèrdua d'informació d'un arxiu (no hem d'oblidar que estarem tractant amb informació de pacients) és per aquest motiu que cap arxiu serà esborrat, simplement serà desplaçat a una altre carpeta un cop haguem acabat de treballar amb ell. Amb aquest tipus d'estructura sempre podrem recuperar el fitxer d'un pacient.

Tota l'aplicació tindrà un seguiment mitjançant logs que escriurem en un fitxer, aquest fitxer serà diari. L'esborrament dels logs el podrem programar.

Interacció amb altres aplicacions

Actualment el l'aplicació deixa els fitxers dcm en la carpeta "C:\ConversorDicom\dcm\electro" i és el programa "MioComm" de l'empresa C2C qui s'encarrega de fer l'storage al PACS.

Disseny de l'aplicació

L'aplicació s'ha dissenyat pensat en una fàcil incorporació de nous fabricants, podem dividir l'aplicació en dos parts, la primera seria comuna a tots els fabricants i la segona seria particular per cada fabricant.

La part particular consistiria en la lectura, la validació, donar el format a les dades obtingudes del fitxer xml i la posterior càrrega de les variables en la classe comuna EstructuraDicom.

La part comuna s'encarrega de la creació del fitxer dcm, aquest fitxer està compost per les variables que hem rebut de l'xml i per variables que són comunes a tots els fitxers dicom; les variables comunes es poden diferenciar en dos tipus, unes que seran constants (constantsDicom) per tots els fitxers i altres que seran els identificadors dels fitxer (UID).

Hi ha una classe central que s'encarrega de determinar les accions que va fent l'aplicació, aquesta classes es EcgToDcm. La classe Main de l'aplicació només s'encarrega de crear un fil d'execució i li dona el control a la classe Dimoni, que és l'encarregada d'evitar que l'aplicació s'aturi.

ESTRUCTURA DEL CONVERTSOR

L'aplicació el podem dividir en dos parts, la primera part estaria formada per totes aquelles classes que serien comuns a qualsevol fabricant i la segona que seria específica per cada fabricant que volgués incorporar-se al projecte.

Part comuna:

Classes:

- ConfigValidate.java:
 - Encarregada de validar que el mètode de validació de rutes i arxius configurats en el fitxer “configuracion.ini” hagi finalitzat correctament. En el cas de que no sigui així s'encarrega de setejar una variable booleana a fals, per tal de que quan el Dimoni validi si te que tornar a executar el mètode 'realizarConversion()' no l'executi.
- ConstantesDicom.java:
 - Conté totes les constants que necessita un fitxer Dicom (ECG General).
- ConversionProcess.java:
 - Valida que l'arxiu entrant sigui xml, en cas afirmatiu començaria una conversió.
- CrearDicom.java:
 - Classe abstracte on es defineixen els mètodes de la classe CrearDicomGeneral amb l'objectiu de poder canviar si fos necessari a altre tipus de prova.
- CrearDicomGeneral.java:
 - Aquí es creen els arxius .dcm i s'insereixen les dades als camps corresponents de Dicom.

- Demonio.java:
 - Encarregada d'evitar que el conversor s'aturi
- EcgToDcm.java:
 - Porta tot el pes de l'aplicació, és la classe central (encarregada de anar cridant les diferents classes i mètodes) .
- EstructuraDicom.java:
 - Conté totes les variables necessàries per crear un fitxer Dicom (Ecg General).
- Hexadecimal.java:
 - Mètodes amb operacions per treballar amb hexadecimals, necessaris per construir les ones.
- Log.java:
 - Ens permet manegar el registre (log) de les diferents operacions que hi ha al programa.
- Main.java:
 - Crear un fil d'execució i li dona el control a la classe Dimoni.
- Nfo.java:
 - Genera fitxer .nfo per poder fer un storage del dicom.
- Uid.java:
 - Conté les operacions necessàries per generar Uid's (identificadors únics) pel fitxer final.
- ValidacionDicom.java:
 - S'encarrega de corroborar que les dades que tenim guardades a la nostra "EstructuraDicom" compleixin els estàndards Dicom segons el Iod que pertoqui. En el cas de que no sigui així el fitxer .xml tractat és mogut a la carpeta d'erronis.

ESTRUCTURA DE CARPETES

L'estructura de carpetes que utilitzaríem seria:

- Xml : carpeta on arribarien els diferents arxius xml que exporta la màquina que fa la prova d'esforç
- XmlProcesados: quan fitxer es transformat correctament a format Dicom, l'xml que havíem rebut el passem a aquesta carpeta.
- XmlErroneos: quan un fitxer Xml no pot ser processat passa a aquesta carpeta.
- Dcm: carpeta on es deixen els arxius dcm creats.
- Logs: conté l'arxiu de configuració log4j.properties (configuració de la sortida que es vols dels diferents logs a les diferents classes), i els diferents logs que es van creant.
- Lib: conté les llibreries necessàries per al projecte.

Arxius de configuració:

- Configuració.ini:
 - arxiu per configurar :
 - la ubicació de l'arxiu de configuració
 - les rutes del programa
 - la ubicació de l'arxiu de configuració dels logs
 - configuració dels Aet (identificadors de la màquina on s'ha fet l'electro)
 - definició del root dels diferents Uid's
 - ubicació dels diferents arxius xsd
 - temps d'execució del conversor
 - temps d'execució per esborrar arxius antics

FIL D'EXECUCIÓ D'UN ARXIU

Per desenvolupar l'aplicació s'ha utilitzat un thread de control Daemon, amb aquesta estructura quan l'aplicació acaba la conversió d'un arxiu espera fins que arriba un altre arxiu per convertir, el mètode run() del thread espera noves peticions per continuar executant-se.

1. Creació d'un objecte "ConversionProcess" i el carreguem a una variable del tipus "ConversionProcess"
2. Creació un objecte log i el carreguem a una variable de tipus Log.
3. Creació del fitxer .ini
4. Escollim el llenguatge dels logs (Cat // Cast)
5. Carreguem l'extensió del tipus d'arxiu que entra (en el nostre cas Xml)
6. Creació i configuració de l'arxiu log
7. Comprovació de l'estructura de carpetes que necessitem
8. Creació d'un filtre d'arxius, per llistar només aquells amb extensió xml
9. Llistem els fitxer amb extensió xml i agafem el primer
10. Escollim un fabricant (actualment només hi ha un: General Electric)
11. Validem el fitxer xml, mitjançant un xsd que hauria de proporcionar el fabricant i mitjançant uns mètodes propis

Nota: En cas que sigues correcta continuaria el procés, en el cas que fos incorrecta el fitxer seria desplaçat a la carpeta XmlErroneos. Validació correcte:

12. Lectura del fitxer xml i carreguem les variables a la classe EstructuraXml

13. Creació del nom del fitxer Dicom

14. Obtenim el nom de la carpeta on es troba l'xml i el carreguem (ens servirà per obtenir la màquina des de on s'ha realitzar la prova d'esforç)

15. Creació de l'objecte Nfo

16. Creació dels mòduls corresponent a un Ecg General

17. Carreguem totes les variables Dicom bolcant la informació rebuda de l'xml (es dona el format correcte a les variables)

18. Creem l'objecte ValidacionDicom i comprovem que l'objecte creat es correcte.

19. Si la validació es correcta creem l'objecte EcgGeneral

20. L'arxiu xml que ha sigut processat es mogut a la carpeta XmlProcesados

21. Creació del fitxer Dicom

22. Afegim els mòdul dicom al fitxer

23. S'escriu el fitxer dicom

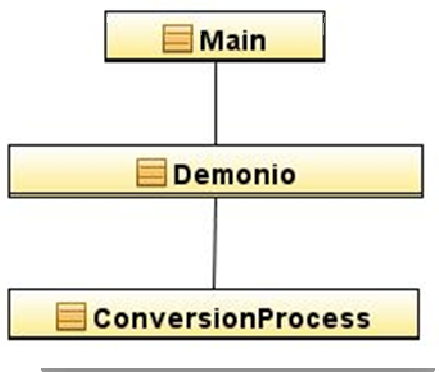
24. Es fa una comprovació per veure si es vol crear l'arxiu .nfo. (Si / No : es llegeix des de l'arxiu de configuració)

25. En cas afirmatiu creem l'arxiu .nfo.

26. Es posen les variables usades a null

DIAGRAMES DE CLASSES

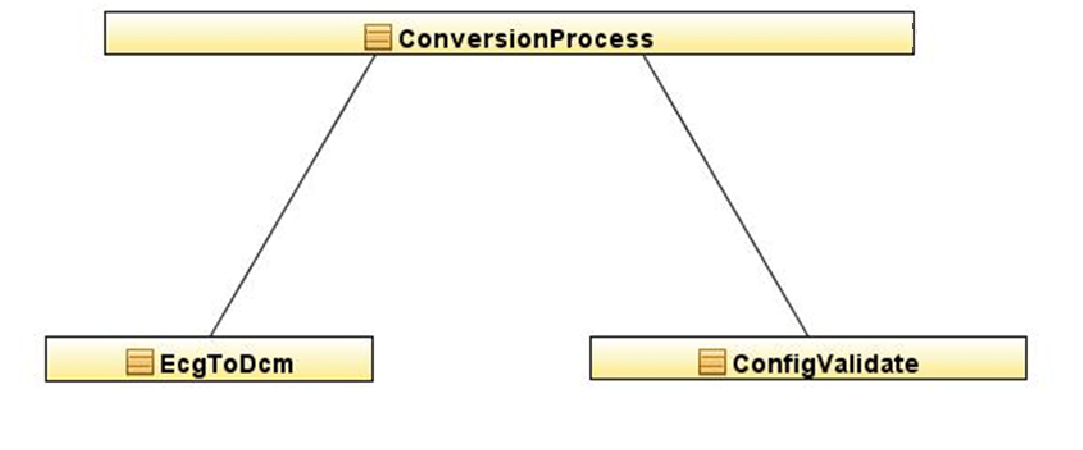
DIAGRAMA DE CLASSES DEMONI



'Main' és la classe principal i la seva única feina es crear un objecte de la classe Demonio. Aquesta classe s'encarrega d'engegar el fil d'execució i gestionar-lo.

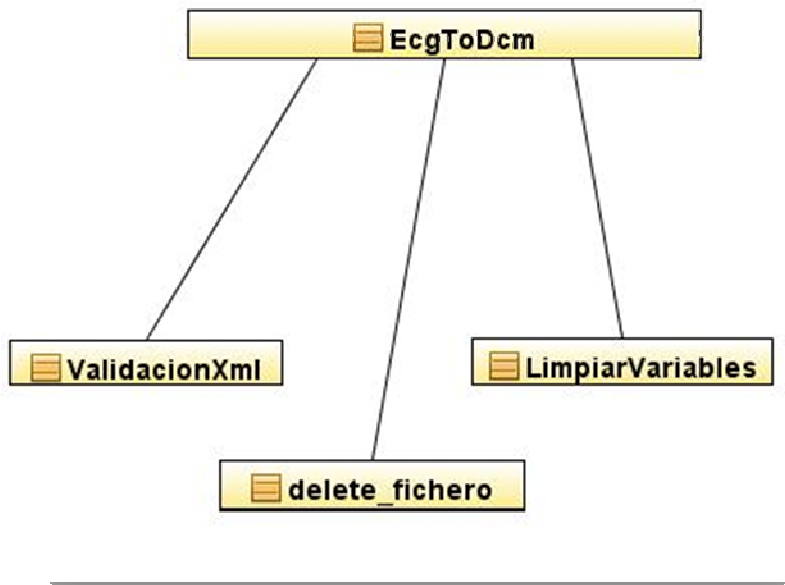
Per a que el conversor es pugui executar indefinidament, la classe 'Demonio' crea un objecte 'ConversionProcces' per tal de trucar al mètode 'realizarConversion()'. La classe 'ConversionProcess' valida que es pugui dur a terme la conversió, i en el cas de que així sigui, la classe 'Demonio' truca al mètode 'realizarConversion()' per a que realitzi tot el procés de transformació d'un objecte .xml a un objecte .dcm. La relació de classes és 1 a 1.

DIAGRAMA DE CLASSES CONVERSION PROCESS



La classe 'ConversionProcess' és l'encarregada de validar que el mètode de validació de rutes i arxius configurats en el fitxer "configuracion.ini" hagi finalitzat correctament. En el cas de que no sigui així s'encarrega de setejar una variable booleana a fals, per tal de que quan el Dimoni validi si te que tornar a executar el mètode 'realizarConversion()' no l'executi.

També conté el mètode 'realizarConversion()' que s'encarrega de cridar al mètode 'conversorEcgToDcm()' de la classe 'EcgToDcm'. El mètode 'realizarConversion()' és el que utilitza el Dimoni per executar el nostre conversor indefinidament. Abans de cada execució valida que el valor introduït a la clau "archivoEntrante" en el fitxer "configuracion.ini" sigui "xml". La relació de classes és 1 a 1.

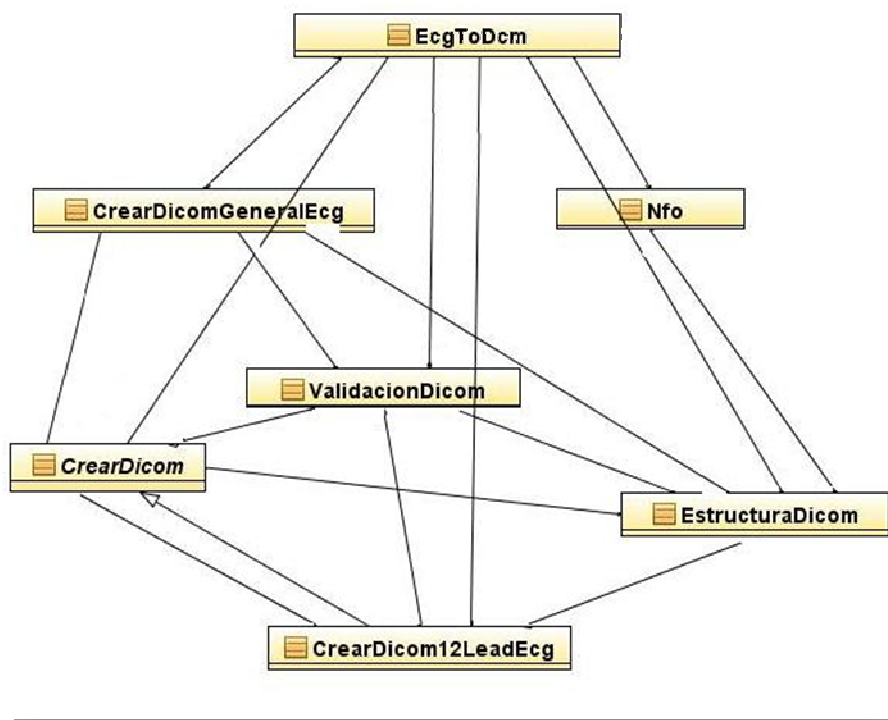
DIAGRAMA DE CLASSES ECG TO DCM : PART 1

La classe "EcgToDcm" és l'encarregada d'organitzar i portar tota l'activitat del conversor a través de les seves diferents crides als mètodes que pertorquin de les diverses classes.

En aquesta primera part del diagrama crida a les classes valida els Xml de General Electric.

També conté un mètode "borraFicheros()" per crear l'objecte i cridar un mètode de la classe "delete_fichero" que serveix per esborrar fitxers que han quedat obsolets. La relació de classes és 1 a 1.

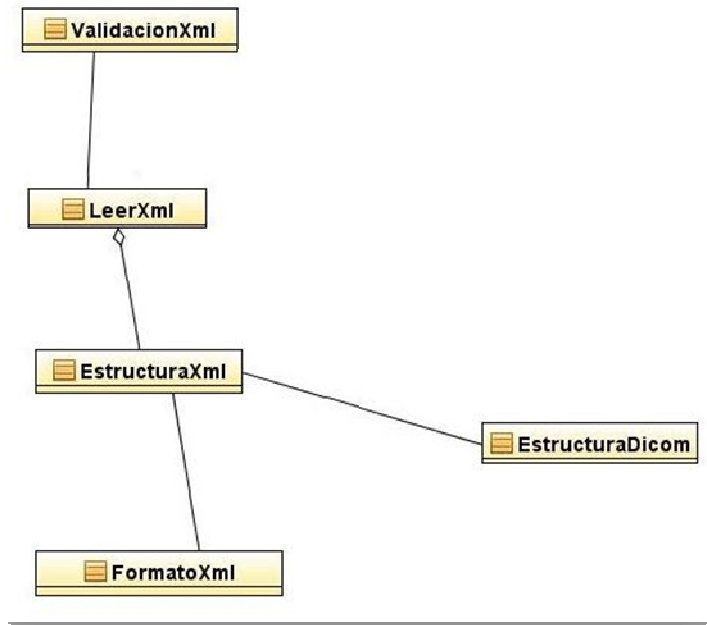
DIAGRAMA DE CLASSES ECG TO DCM: PART 2



En aquesta segona part del diagrama "EcgToDcm" llista tots els fitxers que hi hagin a la carpeta configurada al fitxer de configuració com entrant dels arxius amb extensió .xml. Posteriorment analitza de quin tipus de fabricant i prova es tracta el fitxer que s'ha agafat de la carpeta entrant de xml's. A partir del tipus de fitxer que tinguem es farà el traspàs de dades obtingudes del .xml a l'estructuraDicom.

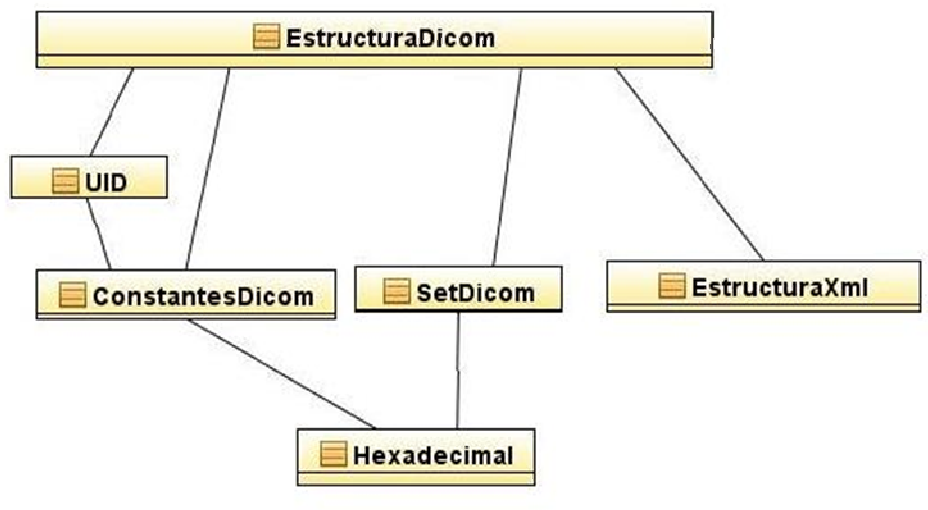
Després es valida que l'estructuraDicom creada i el valor de les seves variables compleixi amb l'estàndard que marca Dicom.

Si tot ha anat correctament procedim a afegir tota la informació que tenim a la EstructuraDicom al objecte Dicom, aquest anirà en una corresponent carpeta la qual tindrà un nom compostat per l'identificació del pacient i l'hora i data del sistema. La relació de classes és 1 a 1.

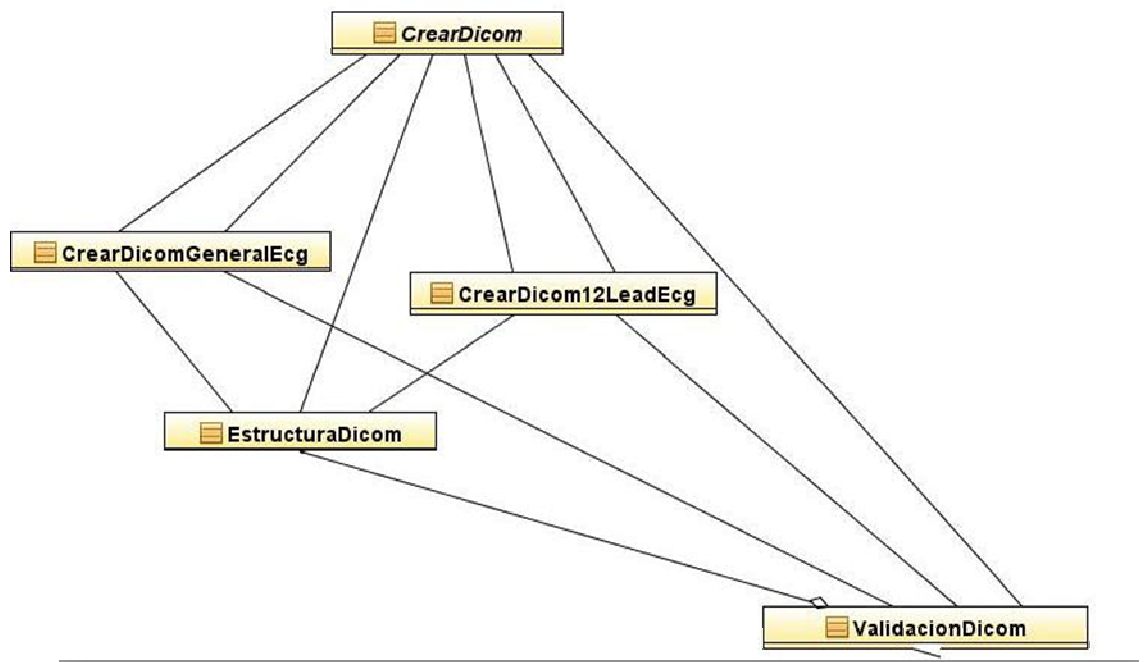
DIAGRAMA DE CLASSES VALIDACIÓ

Aquestes classes s'encarregaran de fer el tractament dels fitxers xml de General Electric, des de l'arribada del fitxer fins l'assignació de les variables necessàries per poder construir un objecte de tipus Dicom. La relació de classes és 1 a 1.

DIAGRAMA DE CLASSES ESTRUCTURA DICOM

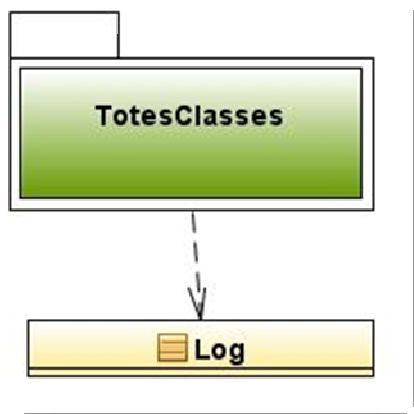


La classe "EstructuraDicom" és l'encarregada de crear els mòduls Waveform i d'inicialitzar els valors constants segons l'electrocardiograma que estem tractant a més d'inicialitzar els valors de les variables UID's i crear l'objecte SetDicom que pertanyi al fabricant del fitxer .xml. La relació de classes és 1 a 1.

DIAGRAMA DE CLASSES CREAR DICOM

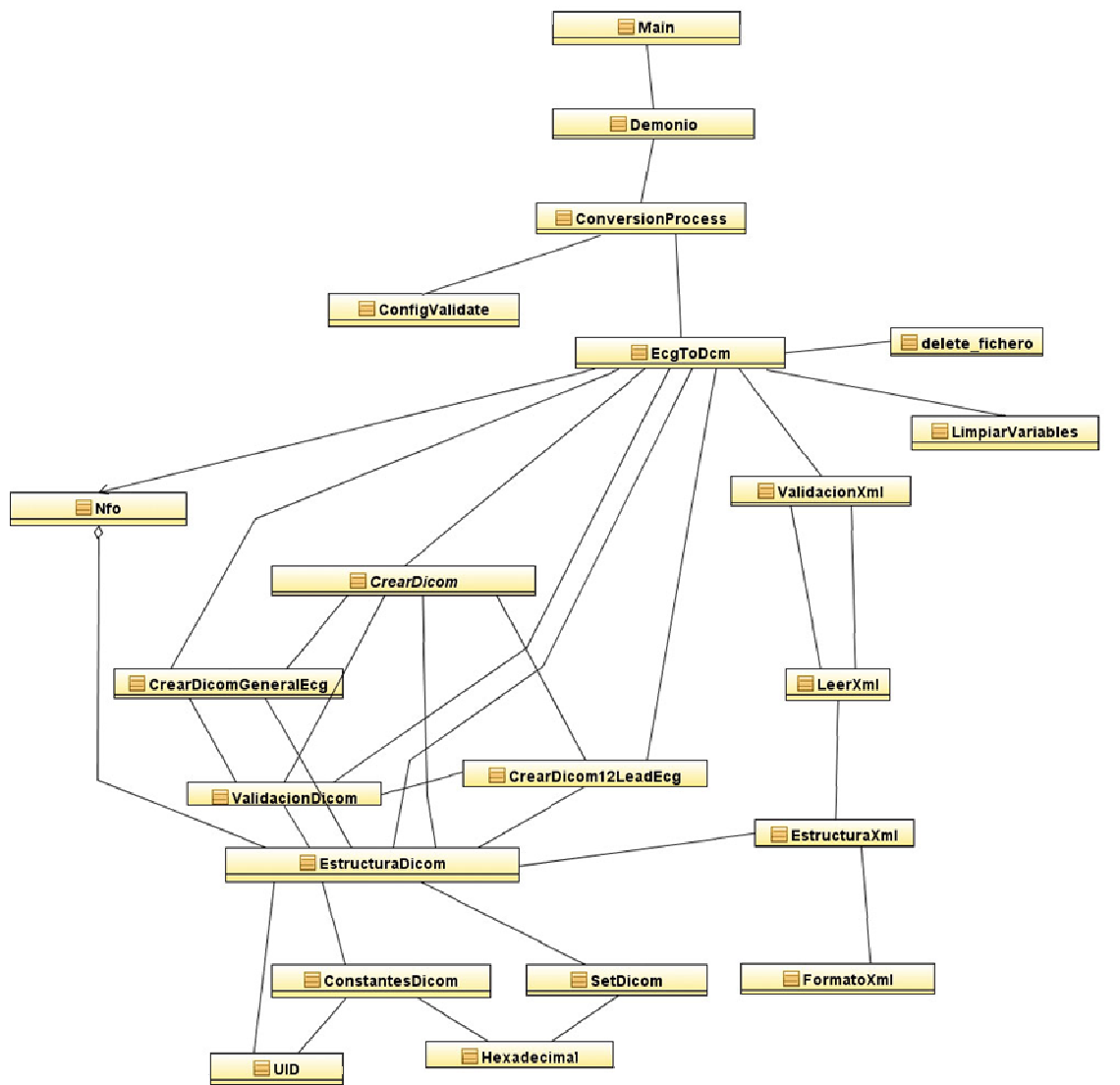
Aquí es creen els arxius .dcm i s'insereixen les dades als camps corresponents de Dicom. La relació de classes és 1 a 1.

DIAGRAMA DE CLASSES LOGS



Aquesta classe està relacionada amb totes les classes.

DIAGRAMA DE CLASSES COMPLERT



DIAGRAMES DE SEQÜÈNCIA

DIAGRAMA DE SEQÜÈNCIA DIMONI

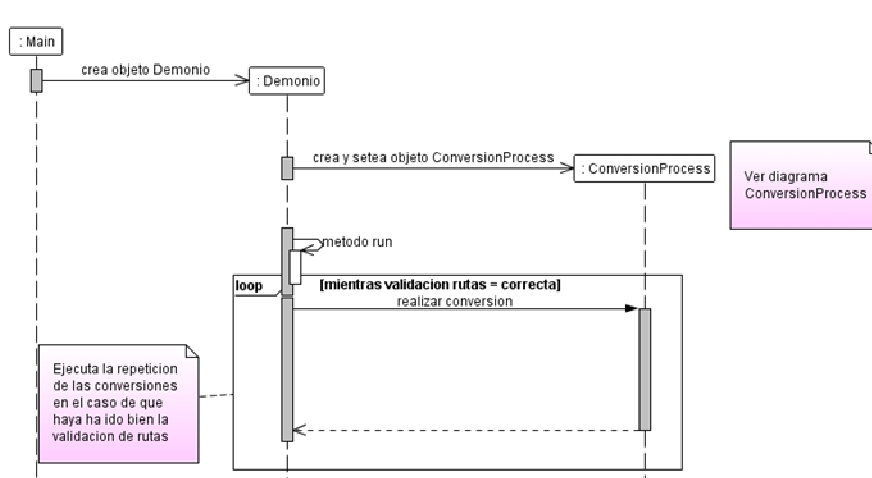


DIAGRAMA DE SEQÜÈNCIA CONVERSION PROCESS

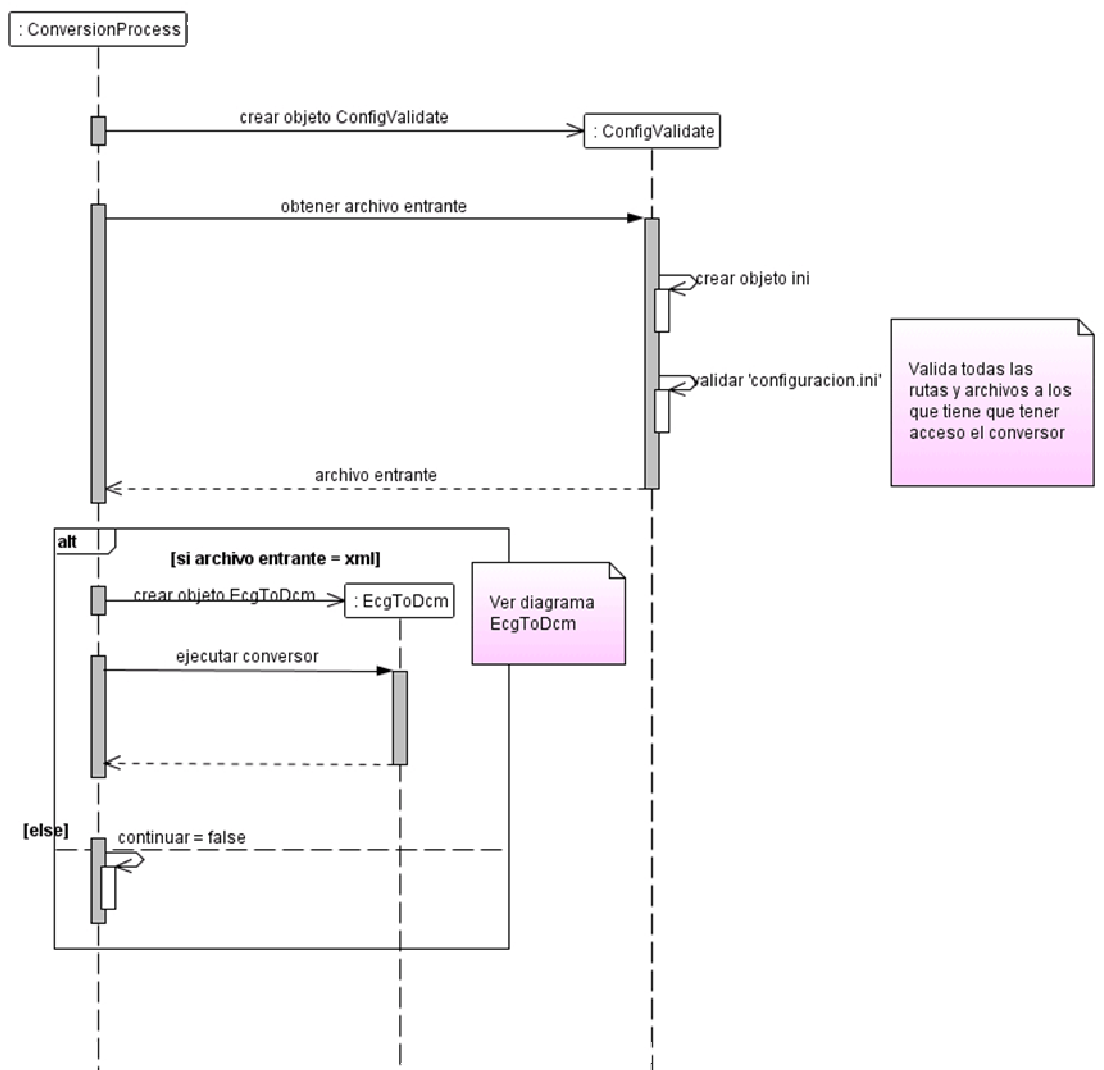


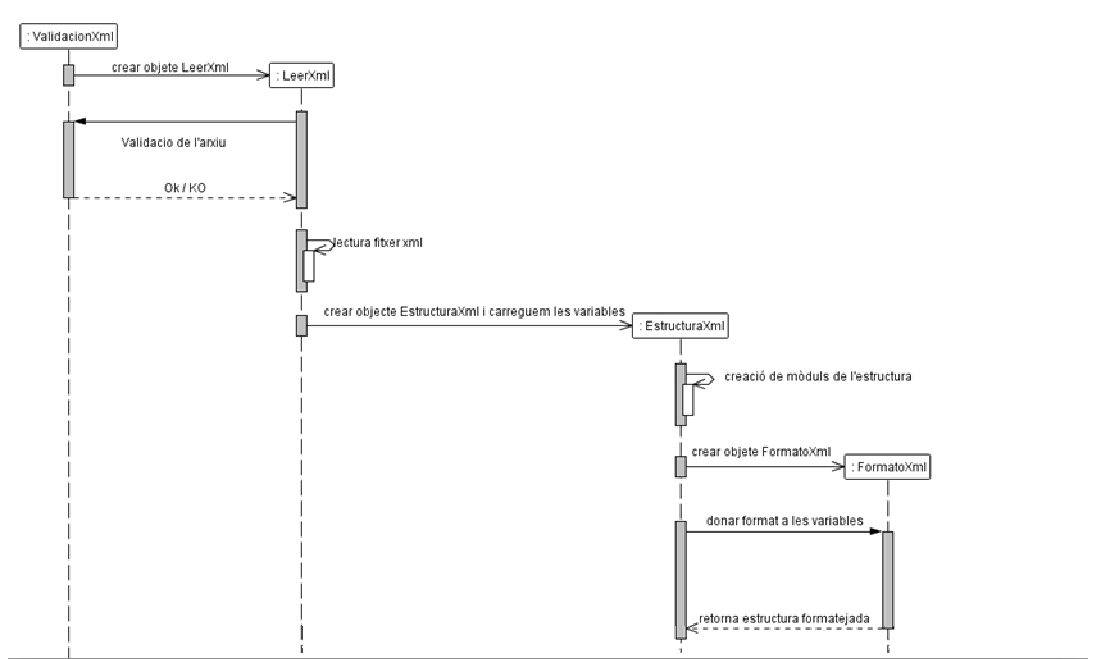
DIAGRAMA DE SEQÜÈNCIA VALIDACIÓ XML

DIAGRAMA DE SEQÜÈNCIA ECG TO DCM (següent pàgina)

Metodo que coge todos los ficheros que haya en la carpeta de localizacion configurada mediante el fichero configuracion.ini y que va realizando todos los pasos de la conversion hasta finalizar.

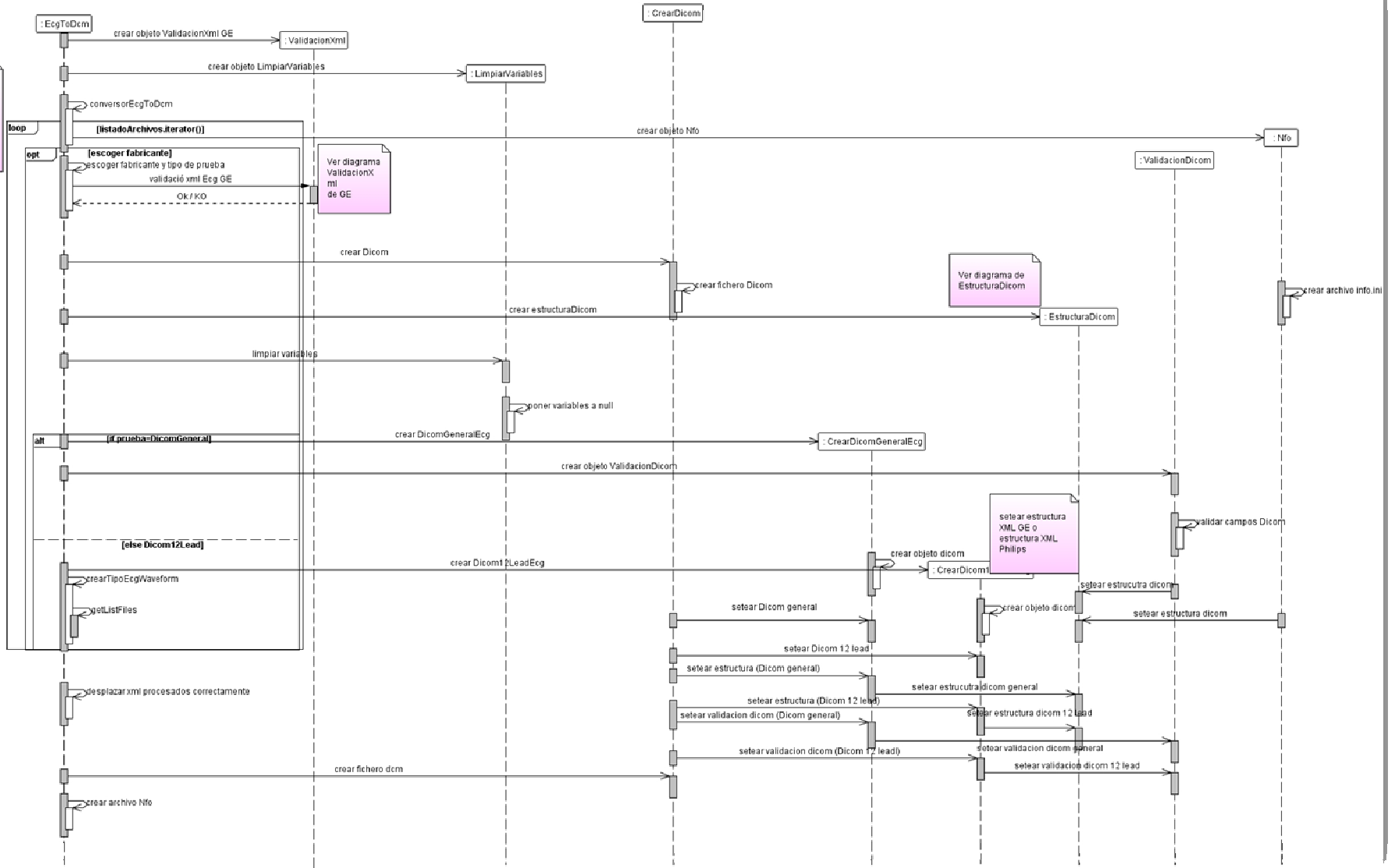
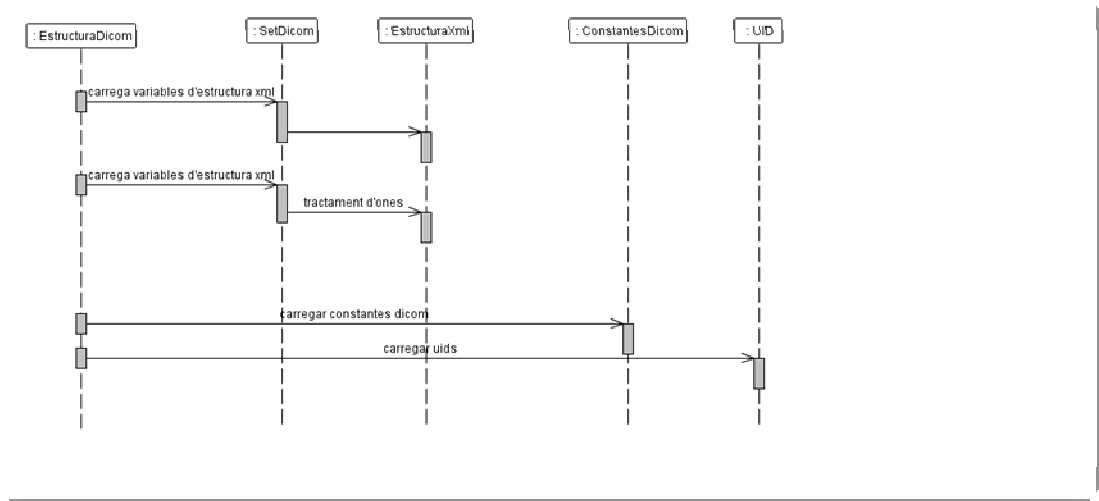


DIAGRAMA DE SEQÜÈNCIA ESTRUCTURA DICOM

Manual d'usuari

Instal·lació del Conversor

1-Copiar l'estructura de carpetes en una unitat

Exemple:

c:\ConversorDicom\xml

c:\ConversorDicom\xmlErroneo

c:\ConversorDicom\xmlProcesado

c:\ConversorDicom\dcm

c:\ConversorDicom\logs

c:\ConversorDicom\lib

c:\ConversorDicom\configuracion

2-Copiar l'executable del conversor al directori arrel de l'aplicació

Exemple:

c:\ConversorDicom\ConversorDicom.exe

3-Copiar l'executable nssm.exe al directori arrel de l'aplicació

4-Configurar l'arxiu "configuracion.ini"

5-Configurar l'arxiu "info.ini" que es troba a la carpeta configuració de l'aplicació

6-Establir el conversor com una aplicació:

Per instal·lar el servei cal anar a la consola de comandos, situar-se a la carpeta on està l'executable nssm.exe i escriure:

```
nssm install <nom de servei> <ruta absoluta de l'aplicació>
```

Requeriments de software:

-Aquesta aplicació funciona en els següents sistemes operatius:

- Windows XP 32 bits

- Windows 2003 server x64 (Standard Edition)

- Windows 2003 server x32 (Standard Edition)

- Windows 2003 server x32 (Enterprise)

- Windows 2003 server x32 (Enterprise)

-JRE (Java Runtime Enviroment) superior 1.4

-Programa “Nssm“ que ens permetrà posar l’aplicació com un servei Windows.

Integració de Fabricants

Fabricants integrats en l'aplicació

En el projecte s'ha incorporat el fabricant General Electrics (GE).

Material proporcionat per GE:

- Diversos fitxers xml de la màquina "MAC1200"
- Estàndard dels atributs xml que exporten les seves màquines.
- Manual de Cardiosoft (software que exporta els fitxers).

Incorporació d'un nou fabricant:

Si un fabricant es vol integrar a l'aplicació hi hauria que afegir un altre paquet al projecte compost per cinc classes: LeerXml, FormatoXml, EstructuraXml, ValidacionXml i SetDicom. Tal i com s'ha dissenyat l'aplicació la resta de classes es reutilitzaran i no serà necessari fer cap canvi.

ESTÀNDARD DICOM

EXPLICACIÓ

Estàndard per la comunicació d'imatges i els formats de missatges per imatges diagnòstiques i terapèutiques.

Establert per ACR (American College of Radiology) y NEMA (National Electrical Manufacturer Association) per facilitar la interoperabilitat entre dispositius.

En el seu desenvolupament s'ha mantingut un alt grau de col·laboració amb altres organitzacions d'estandardització incloent CENT TC251 en Europa i JIRA en Japó, participant en la revisió també altres organitzacions d'Estats Units com IEEE, HL7 i ANSI.

Està basat en altres estàndards com Ethernet, TCP/IP, arquitectures Client/Servidor i la programació orientada a objectes .

DICOM està suportat per la gran majoria de fabricants d'equipament radiològic i de PACS (Picture Archiving and Communications Systems).

Aquest estàndard s'ha incorporat en la norma Europea MEDICOM (Medical Image Communication)

DICOM VERSIÓ 3.0

L'estàndard actual DICOM versió 3.0 recull un gran nombre de millores en relació amb les versions anteriors que només eren aplicables per sistemes de comunicació punt a punt.

DICOM versió 3.0 suporta el treball en xarxa amb protocols estàndard de tipus OSI i TCP/IP.

DICOM versió 3.0 descriu explícitament com un tècnic té que implementar una declaració de conformitat ("Conformance Statement") per seleccionar ordres específiques. També introdueix objectes d'informació no només per les imatges i els gràfics sinó també per els estudis, els informes...

L'estàndard DICOM versió 3.0 facilita la interoperabilitat dels equips d'imatges mèdiques especificant:

- Un conjunt de protocols que tenen que seguir els dispositius.
- Una sintaxis i una semàntica de comandos i la informació associada que es pot intercanviar usant aquests protocols.

La norma DICOM no especifica:

- Els detalls de la implementació de qualsevol característica de l'estàndard en un equip.
- El conjunt general de característiques i funcions que s'espera que compleixi un sistema integrat per un grup de dispositius tipus DICOM.
- Un procediment de validació i prova per avaluar el compliment de la norma per un equip o sistema.

S'ha de tenir en compta que encara que dos fabricant assegurin que compleixen l'estàndard DICOM això no assegura la interoperabilitat entre ells.

Els fitxer DICOM consisteixen en dos parts diferenciades:

- Una capçalera amb multitud de camps estandarditzats que especifiquen tant les dades administratives (dades dels pacients, hospital on es va realitzar,...) com dades de la imatge.
- El cos de la imatge que pot estar comprimida amb varis estàndards (JPG, PDF...), seria una encapsulació.

DICOM ha sigut un èxit des del punt de vista comercial i de l'usuari.

ESTRUCTURA DELS FITXERS DICOM

Els fitxers DICOM són el resultat d'un procés d'arxivar imatges i per tant provenen d'un missatge DICOM. Aquest missatge està format per una capçalera i un compost IOD. Per generar el fitxer DICOM és necessari afegir al IOD una nova capçalera anomenada *File Meta Information*.

El format de fitxer DICOM és molt complex, a causa de la gran quantitat de camps que s'especifiquen a la capçalera, així com els diversos tipus de capçalera que permet i la multitud de formats en els quals hi pot haver gravada la imatge.

Des del punt de vista del tècnic, un fitxer DICOM es pot dividir en quatre parts diferenciades:

1. Preàmbul i prefix identificatiu del fitxer.
2. Metacapçalera.
3. Capçalera.
4. Imatge (encara que des del punt de vista del format, la imatge és un element més de la capçalera).

PREÀMBUL

L'estàndard DICOM especifica que un fitxer en format DICOM ha de començar amb un preàmbul.

Aquest preàmbul té una mida fixa de 128 bytes, i està pensat per tenir un ús definit per la implementació. Per exemple, pot contenir informació sobre el nom de l'aplicació usada per crear el fitxer, o pot tenir informació que permeti a aplicacions accedir directament a les dades de la imatge emmagatzemada al fitxer.

L'estàndard DICOM no especifica la manera en la qual les dades del preàmbul hagin de ser estructurades, sent això decisió dels encarregats de dissenyar la implementació.

En cas de no ser usat, el preàmbul ha de ser present, amb tots els seus bytes posats al valor 00h.

Exemple d'estructura:

bytes 0-127 llibres
bytes 128-131 "DICM" (firma DICOM)
metadata (codificat en ELE)
(0002,0000) File Meta Information Group Length UL
1
(0002,0001) File Meta Information Version OB 1
(0002,0002) Media Storage SOP Class UID UI 1
(0002,0003) Media Storage SOP Instance UID UI 1
(0002,0010) Transfer Syntax UID UI 1
(0002,0012) Implementation Class UID UI 1
(0002,0013) Implementation Version Name SH 1
(0002,0016) Source Application Entity Title AE 1
(0002,0100) Private Information Creator UID UI 1
(0002,0102) Private Information OB 1

PREFIX

El que segueix al preàmbul és el prefix identificatiu dels fitxers DICOM.

Aquest prefix consisteix en quatre bytes que contenen la cadena de caràcters DICM. Aquesta cadena ha d'estar codificada sempre amb les lletres en majúscula i usant el repertori de caràcters ISO 8859 G0.

El propòsit d'aquest prefix és permetre a les implementacions diferenciar si un fitxer és DICOM o no.

ELEMENTS DE DADES

La capçalera i la metacapçalera d'un fitxer DICOM consisteixen en una sèrie de camps amb tota la informació necessària sobre la imatge en qüestió, incloent la pròpia imatge.

Entre aquests camps es troben, per exemple, dades sobre el pacient (nom, sexe), sobre el tipus d'imatge i molts més. Entre tots aquests camps els que més ens interessin per al projecte són els que contenen informació necessària per processar i visualitzar la imatge correctament.

Al conjunt de tota la informació codificada sobre un camp se li coneix amb el nom d'Element de Dades (Data Element). Així, tant la capçalera com la metacapçalera d'un fitxer DICOM consisteixen en una successió d'elements de dades.

Un element de dades està constituït pels camps:

- Etiqueta de l'Element de Dades (Data Element Tag):

Serveix per identificar cada element de dades de forma unívoca. Aquesta etiqueta està constituïda per un Nombre de Grup (Group Number) i un Número|Nombre d'Element (Element Number). Al document número 6 de l'estàndard hi ha les descripcions de tots els elements de dades, ordenades segons aquesta etiqueta. Al document número 3 de l'estàndard s'explica el propòsit de cada un d'ells, així com la seva obligatorietat. Exemple, si el número|nombre de grup són vuit i el número|nombre d'element són dotze, l'etiqueta serà (0008,000C).

- Representació del Valor (Value Representation, VR abreujat):

Indica la forma en la qual es codifica el valor de l'element. Per exemple el valor pot estar codificat com una cadena de caràcters o un enter sense signe. Més endavant donarem el llistat amb tots els possibles VRs. El camp VR no sempre està codificat en l'element de dades, sinó que depèn de la sintaxi de transferència.

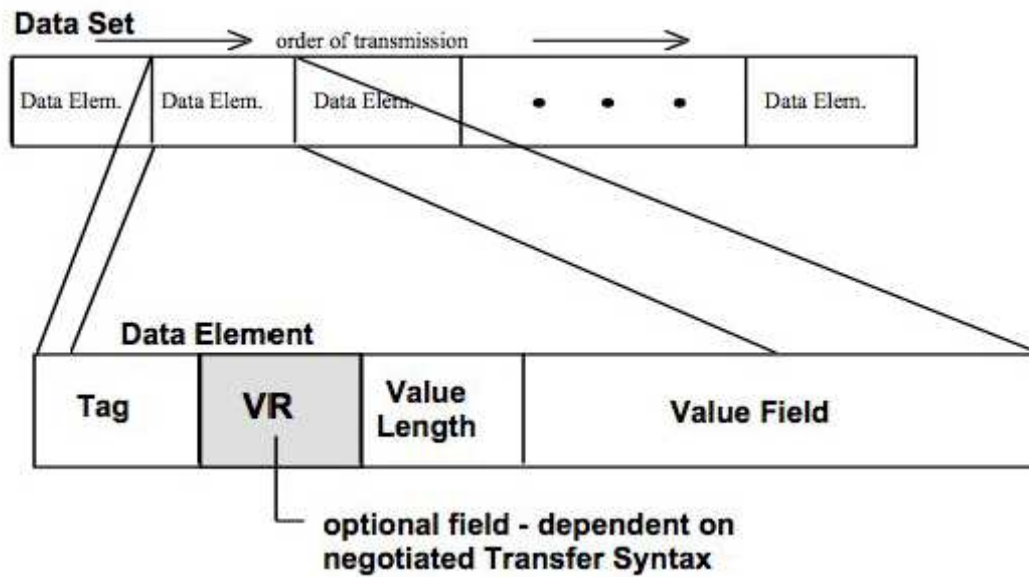
- Longitud del Valor (Value Length):

Com el seu nom indica, és la longitud del camp Valor.

- Valor (Value):

És el valor de l'element de dades, codificat segons el camp VR i amb la longitud que indica el camp Longitud del Valor. Per exemple, per a l'element de dades Nom del pacient (0010,0010) el valor podria ser "Javier Pérez", amb una longitud del valor igual a deu i amb la representació de valor "PN" (Person Name).

Exemple:



Camps DICOM

Els camps d'una capçalera i metacapçalera DICOM contenen, per una part, tota la informació necessària perquè una implementació de l'estàndard sigui capaç de processar i visualitzar correctament la imatge o imatges emmagatzemades en un fitxer DICOM, i, d'altra banda, totes les dades associades a aquestes que el personal mèdic necessita per interpretar correctament la imatge o imatges en qüestió.

Tots els camps definits per DICOM es troben llistats en una base de dades que es troba al document 6 de l'estàndard, i se la coneix com a Registre dels elements de dades DICOM (Registry of DICOM data elements).

Cada element de dades està indexat per la seva etiqueta (nombre de grup i número|nombre d'element), que és la clau usada per buscar-ho en la base de dades, i per a cada element de dades se'ns especifica:

Nom: nom de l'element de dades, que és una petita descripció de la seva funció.

VR: representació del valor de l'element. En cas que la capçalera sigui de tipus VR implícita, serà necessari consultar aquesta base de dades per conèixer el seu VR.

Digitalització Imatge Mèdica no Radiològica

VM: indica la quantitat de valors del mateix tipus que poden contenir el camp Valor de l'element de dades. Per exemple l'element de dades (0010,2154) Números|Nombres de Telèfon del Pacient (Patients Telephone Numbers) pot contenir més d'un

número|nombre de telèfon del pacient, mentre que l'element de dades (0028,0004) Interpretació Fotomètrica (Photometric Interpretation) sol pot tenir un valor, ja que una imatge no pot tenir més d'una interpretació fotomètrica.

Si l'element de dades és obsolet i ha estat retirat en la versió actual de l'estàndard, tindrà l'identificador RET.

D'altra banda, al document 3 de l'estàndard vénen tots els camps explicats detalladament. Per a cada un d'ells, se'ns especifica la seva funció, la forma en la qual s'ha de codificar el seu valor i la seva obligatorietat d'ús.

CODIFICACIÓ D'UN FITXER DICOM

Existeixen dos tipus de codificació per els elements de dades:

- VR explícita
- VR implícita

1-Codificació VR Explícita

En aquesta codificació, els elements de dades es codifiquen de dues formes diferents, depenent del seu VR.

Si la VR de l'element de dades és OB, OW, OF, SQ, UT o UN, la codificació és la següent:

1. L'etiqueta es codifica usant quatre bytes: els dos primers per al número|nombre de grup i els dos últims per al número|nombre d'element. És a dir, una parella d'enters sense signe de 16 bits.

2. La VR es codifica com una cadena de caràcters de dos bytes, que pot tenir el valor OB, OW, OF, SQ, UT o UN.

3. Els dos bytes següents són reservats, i han de ser posats al valor 0000h.

4. Els següents quatre bytes són per a la longitud del valor, codificada com un enter sense signe de 32 bits.

5. Els següents bytes, tants com indica la longitud del valor, -llevat del cas especial que veurem més endavant, són per al valor, que estarà codificat segons indica el seu VR i la sintaxi de transferència.

Si la VR de l'element de dades és qualsevol que no sigui una de les anteriorment esmentades, la codificació és la següent:

1. L'etiqueta es codifica usant quatre bytes: els dos primers per al número|nombre de grup i els dos últims per al número|nombre d'element. És a dir, una parella d'enters sense signe de 16 bits.

2. La VR es codifica com una cadena de caràcters de dos bytes, que pot ser qualsevol VR menys les esmentades anteriorment.

3. Els següents dos bytes són per a la longitud del valor, codificada com un enter sense signe de 16 bits.

4. Els següents bytes, tants com indica la longitud del valor, són per al valor, que estarà codificat segons indica el seu VR i la sintaxi de transferència.

2-Codificació VR Implícita

Aquesta és la codificació que s'usa per a la sintaxi de transferència per defecte, i la diferència més important respecte amb l'anterior (VR explícita) és que, com el seu nom indica, la VR de cada element de dades no es codifica al fitxer, obligant a consultar els documents de l'estàndard (o una base de dades externa al fitxer) per obtenir-la, igual com ocorre amb la descripció de l'element de dades i amb la seva obligatorietat.

La codificació és la següent:

1. L'etiqueta es codifica usant quatre bytes: els dos primers per al número|nombre de grup i els dos últims per al número|nombre d'element. És a dir, una parella d'enters sense signe de 16 bits.
2. Els següents quatre bytes són per a la longitud del valor, codificada com un enter sense signe de 32 bits.
3. Els següents bytes, tants com indica la longitud del valor -llevat d'en el cas especial, són per al valor, que estarà codificat segons indica el seu VR i la sintaxi de transferència.

LLIBRERIES DICOM

LLIBRERIES CONEGUDES

- DcmTk
 - Llenguatge de programació: C, C++
 - Descarregar: <http://dicom.offis.de/dcmTk.php.en>
 - Documentació: <http://support.dcmTk.org/docs/>
 - Llicència: free

- Dcm4che1
 - Llenguatge de programació: JAVA
 - Descarregar: <http://sourceforge.net/project/showfiles.php?>
 - Documentació: accedint a la pantalla de comandos
 - Llicència: free

Utilitzada en el projecte

- Dcm4che2
 - Llenguatge de programació: JAVA
 - Descarregar: <http://sourceforge.net/project/showfiles.php?>
 - Documentació:
<http://www.dcm4che.org/confluence/display/d2/dcm4che2+DICOM+Toolkit>
 - Llicència: free

DCM4CHEE

EXPLICACIÓ DE LA LLIBRERIA

Dcm4che és una col·lecció d'aplicacions de codi obert i utilitats per aplicacions sanitàries. Aquestes aplicacions s'han desenvolupat en el llenguatge de programació de Java. El kit de desenvolupament té que ser igual o superior al JDK 1.4.

Dcm4che és una aplicació robusta que compleix amb les normes DICOM. El joc d'eines de DICOM de dcm4che-1.x s'utilitza en moltes aplicacions de producció a través del món, mentre que la versió actual (2.x) del joc d'eines s'ha desenvolupat per proporcionar alt rendiment i flexibilitat.

Dcm4chee proporciona els següents serveis:

DICOM Storage: Ac dcm4chee pot emmagatzemar qualsevol tipus d'objecte DICOM en un sistema d'arxius estàndard que segueixi les normes DICOM (si fos necessari es podria comprimir l'arxiu que s'envia).

DICOM Query/Retrieve: demanar / recuperar un objecte dicom.

WADO and RID: accés web al contingut arxivat.

Altres serveis DICOM: MPPS, GPWL, MWL, Storage Commitment, Instance Availability Notification, Study Content Notification, Output Content to CD Media, Hanging Protocols....

HL7 Server: funcionaria com un servidor de HL7 amb la possibilitat d'actuar sobre missatger de tipus: ADT, ORM i ORU

Viabilitat implementació conversor mitjançant XSLT:

Actualment la classe que dirigeix les operacions del conversor (guia tot el procés) és *EcgToDcm*, aquesta classe continuaria sent el nucli del programa, les modificacions vindrien posteriorment, aquesta classe en comptes de cridar la classe *leerXml* (s'encarrega de llegir l'arxiu i carregar les variables a la classe *EstructuraXml*) cridaria al fitxer XSLT d'un fabricant determinat (aquesta classe llegiria l'xml i agafaria els tags que es decidís), posteriorment es cridaria a la classe *leerXml* i es carregarien les variables a la classe *Estructura xml*, on se li donarien un format correcte, posteriorment el flux de seqüència seria igual que el que hi ha ara fins a la creació del fitxer DICOM.

Millores : actualment per cada fabricant s'han de desenvolupar 5 classes (cada mòdul: *leerXml*, *EstructuraXml*, *ValidacionXml*, *FormatoXml*, *SetDicom*), amb l'utilització d'XSLT només seria necessari incorporar codi per cridar als fitxers *xslt*, ja que tots els fitxers *xml* que es crearan tindran una estructura similar, per tant la lectura, el canvi de format de les variables i la carrega de les variables passarà a ser comuna per tots els fabricants (mitjançant els *xslt* es també faria la validació).

Model de dades DICOM

Tenint en compte tota la informació de l'estàndard DICOM proporcionada anteriorment, aquest seria el model de dades d'un fitxer Dicom per una prova d'esforç que està basat en el model "*TwelveECGWaveformStorage*".

Tag	Atribut DICOM
(0002,0002)	Media Storage SOP Class UID
(0002,0003)	Media Storage SOP Instance UID
(0002,0010)	Transfer Syntax UID
(0002,0012)	Implementation Class UID
(0002,0013)	Implementation Class UID
(0002,0016)	Source Application Entity Title
Tag	Atribut DICOM
(0008,0012)	Instance Creation Date
(0008,0013)	Instance Creation Time
(0008,0016)	SOP Class UID
(0008,0018)	SOP Instance UID
(0008,0020)	Study Date
(0008,0023)	Content Date
(0008,002A)	Acquisition DateTime
(0008,0030)	Study Time
(0008,0033)	Content Time
(0008,0050)	Accession Number
(0008,0060)	Modality
(0008,0070)	Manufacturer
(0008,0080)	Institution Name
(0008,0081)	Institution Address
(0008,0090)	Referring Physician's Name
(0008,1030)	Study Description
(0008,103e)	Series Description
(0008,1048)	Physician(s) of Record
(0008,1060)	Name of Physician(s) Reading Study
(0008,1070)	Technician
(0008,1090)	Manufacturer's Model Name
(0010,0010)	Patient's Name
(0010,0020)	Patient ID

(0010,0030)	Patient's Birth Date
(0010,0040)	Patient's Sex
(0010,1010)	Patient's Age
(0010,1020)	Patient's Size
(0010,1030)	Patient's Weight
(0010,2160)	Ethnic Group
(0010,21B0)	Additional Patient History
(0018,1020)	Software Versions
(0020,000D)	Study Instance UID8212911.109.825761494
(0020,000E)	Series Instance
(0020,0010)	Study ID
(0020,0011)	Series Number
(0020,0013)	Instance Number
(0038,0010)	Admission ID
(0038,0300)	Current Patient Location
(0038,0400)	Patient's Institution Residence
(003A,0210)	Channel Sensitivity
(003A,0211)	Channel Sensitivity Units Sequence
(0008,0100)	Code Value
(0008,0102)	Coding Scheme Designator
(0008,0103)	Coding Scheme Version
(0008,0104)	Code Meaning

Tag	Atribut DICOM
0040,3001)	Confidentiality Constraint on Patient Data Description
0040,0555)	AcquisitionContextSequence
0040,A040)	ValueType
0040,A043)	<u>ConceptNameCodeSequence</u>

(0008,0100)	CodeValue
(0008,0102)	CodeSchemeDesignator
(0008,0103)	CodeSchemeVersion
(0008,0104)	CodeMeaning
(0040,0168)	ConceptCodeSequence
(0008,0100)	CodeValue
(0008,0102)	CodeSchemeDesignator
(0008,0103)	CodeSchemeVersion
(0008,0104)	CodeMeaning

Tag	Atribut DICOM
(0040,B020)	Waveform Annotation Sequence
(0070,0006)	Unformatted Text Value
(0040,A0B0)	Referenced Waveform Channels
(0040,A180)	Annotation Group Number

Tag	Atribut DICOM
(5400,0100)	WaveformSequence
(0018,1068)	MultiplexGroupTimeOffset
(0018,1069)	TriggerTimeOffset
(003A,0004)	WaveformOriginality
(003A,0005)	NumberOfWaveformChannels
(003A,0010)	NumberOfWaveformSamples
(003A,001A)	SamplingFrequency
(003A,0020)	MultiplexGroupLabel

(003A,0200)	ChannelDefinitionSequence
	ITEMs 3
(003A,0208)	>ChannelSourceSequence 1 ITEM
(0008,0100)	>>CodeValue
(0008,0102)	>>CodeSchemeDesignator
(0008,0103)	>>CodeSchemeVersion
(0008,0104)	>>CodeMeaning
(003A,0210)	>ChannelSensitivity
(003A,0211)	>ChannelSensitivityUnits Sequence
(0008,0100)	>>CodeValue
(0008,0102)	>>CodeSchemeDesignator
(0008,0103)	>>CodeSchemeVersion
(0008,0104)	>>CodeMeaning
(003A,0212)	>ChannelSensitivityCorrectionFactor
(003A,0213)	>ChannelBaseline
(003A,0215)	>ChannelSampleSkew
(003A,021A)	>WaveformBitsStored
(003A,0220)	>FilterLowFrequency
(003A,0221)	>FilterHightFrequency
(003A,0222)	>NotchFilterFrequency
(5400,1004)	WaveformBitsAllocated
(5400,1006)	WaveformSampleInterpretation
(5400,100A)	waveformPaddingValue
(5400,1010)	WaveformData

Equivalència Variables Xml de GE amb Tags Dicom

Atributo -FILE FORMAT-	Tag	Valor/campo XML-GE
MediaStorageSOPClassUID	(0002,0002)	Storage Waveform
MediaStorageSOPInstanceUID	(0002,0003)	Ej: 2.16.840.1.113883.4.292.30.14.1.2009110 8212911.109.825761494.1.1
TransferSyntaxUID	(0002,0010)	
ImplementationClassUID	(0002,0012)	Root + IdApplication:
ImplementationVersionName	(0002,0013)	
SourceApplicationEntityTittle	(0002,0016)	
Atributo -DATASET-	Tag	Valor/campo XML-GE
InstanceCreationDate	(0008,0012)	Fecha del envío al PACS
InstanceCreationTime	(0008,0013)	Hora del envío al PACS
SOPClassUID	(0008,0016)	GeneralECGWaveformStorage: 1.2.840.10008.5.1.4.1.1.9.1.2
SOPInstanceUID	(0008,0018)	Ej: 2.16.840.1.113883.4.292.30.14.1.2009110 8212911.109.825761494.1.1
StudyDate	(0008,0020)	Year, Month, Day
ContentDate	(0008,0023)	Year, Month, Day
AcquisitionDateTime	(0008,002A)	Year, Month, Day, Hour, Minute, Second
StudyTime	(0008,0030)	Hour,Minute,Second
ContentTime	(0008,0033)	Hour, Minute, Second
AccessionNumber	(0008,0050)	<OrderNumber>
Modality	(0008,0060)	"ECG"
Manufacturer	(0008,0070)	"GE Medical Systems"
InstitutionName	(0008,0080)	PatientVisit ServicingFacility Name
InstitutionAdress	(0008,0081)	PatientVisit ServicingFacility Adress Street/ City
ReferringPhysiciansName	(0008,0090)	PatientVisit OrderingProvider FamilyName / GivenName
StudyDescription	(0008,1030)	ECG en Repós (RestECG)
SeriesDescription	(0008,103e)	ECG en Repós (RestECG)

PhysiciansOfRecord	(0008,1048)	PatientVisit AttendingDoctor FamilyName / GivenName
NameOfPhysiciansReadingStudy	(0008,1060)	PatientVisit Referring Doctor FamilyName / GivenName
OperatorsName	(0008,1070)	Technician Family/Given Name
ManufacturersModelName	(0008,1090)	Device-Type
Patient'sName	(0010,0010)	PatientInformation PatientName FamilyName / GivenName
PatientId	(0010,0020)	PatientInformation PatientId PID
PatientsBirthDate	(0010,0030)	PatientInformation Birth Date/Time Day / Month / Year
PatientsSex	(0010,0040)	PatientInformation Gender
PatientsAge	(0010,1010)	PatientInformation Age
PatientsSize (cm)	(0010,1020)	PatientInformation Height
PatientsWeigth (kg)	(0010,1030)	PatientInformation Weight
EthnicGroup	(0010,2160)	PatientInformation Race
AdditionalPatientHistory	(0010,21B0)	PatientVisit Medical History MedicalHistoryText
SoftwareVersion	(0018,1020)	DeviceInfo Desc /SoftwareVersion / AnalysisVersion
<u>StudyInstanceUID</u>	(0020,000D)	Ej: 2.16.840.1.113883.4.292.30.14.1.2009110 8212911.109.825761494
<u>SeriesInstanceUID</u>	(0020,000E)	Ej: 2.16.840.1.113883.4.292.30.14.1.2009110 8212911.109.825761494.1
StudyId	(0020,0010)	PatientVisit OrderNumber
SeriesNumber	(0020,0011)	"1"
<u>InstanceNumber</u>	(0020,0013)	"1"
AdmissionId	(0038,0010)	"ROUT"
CurrentPatientLocation	(0038,0300)	PatientVisit AssignedPatientLocation Facility / LocationNumber / LocationName
Patient's Institution Residence	(0038,0400)	PatientVisit PatientRoom

ChannelSensitivity	(003A,0210)	StripData Resolution
ChannelSensitivityUnits Sequence 1 ITEM	(003A,0211)	
CodeValue	(0008,0100)	"uV"
CodeSchemeDesignator	(0008,0102)	"UCUM"
CodeSchemeVersion	(0008,0103)	"1.4"
CodeMeaning	(0008,0104)	"microvolt"

Atributo -CONTEXT-	Tag	Valor/Campo XML-GE
Confidentiality Constraint on Patient Data Description	(0040,3001)	Simulated Patient Data
AcquisitionContextSequence	(0040,0555)	
Interpretation Statements 1 Item por cada interpretación		
ITEM1		
ValueType	(0040,A040)	"CODE"
ConceptNameCodeSequence	(0040,A043)	
CodeValue	(0008,0100)	"5.4.5-33-1"
CodeSchemeDesignator	(0008,0102)	"SCPECG"
CodeSchemeVersion	(0008,0103)	"1.3"
CodeMeaning	(0008,0104)	"Electrode Placement"

Equivalència Variables Xml de GE amb Tags Dicom

Atributo -ANNOTATION-	Tag	Valor/campo XML-GE
Waveform Annotation Sequence	0040,B020	
Interpretation Statements Annotation Group Number = 0 1 Item por cada interpretaci3n (4 items)		
Item1..4<Unformatted Text Value	0070,0006	Diagnostics <Exercise interpretation>
Referenced Waveform Channels	0040,A0B0	"0001 0000"
<Annotation Group Number	0040,A180	"1"

Atributo -WAVEFORM-	Tag	Valor/campo XML-GE
<u>WaveformSequence</u> 3 ITEM	(5400,0100)	
MultiplexGroupTimeOffset	(0018,1068)	"0.0"
TriggerTimeOffset	(0018,1069)	"0.0"
<u>WaveformOriginality</u>	(003A,0004)	Ritmo: "ORIGINAL"
<u>NumberOfWaveformChannels</u>	(003A,0005)	Ritmo: StripData
<u>NumberOfWaveformSamples</u>	(003A,0010)	Ritmo: StripData
<u>SamplingFrequency</u>	(003A,001A)	Ritmo: StripData

		SampleRate
MultiplexGroupLabel	(003A,0020)	Ritmo: "RHYTHM"
<u>ChannelDefinitionSequence</u>	(003A,0200)	3 ITEMS, uno por cada derivación
ITEMs 3		
>ChannelSourceSequence	(003A,0208)	
1 ITEM		
>>CodeValue	(0008,0100)	Leads V2 = "5.6.3-9-4"; Leads V4 = "5.6.3-9-6"; Leads V6 = "5.6.3-9-8";
>>CodeSchemeDesignator	(0008,0102)	"SCPECG"
>>CodeSchemeVersion	(0008,0103)	"1.3"
>>CodeMeaning	(0008,0104)	Lead V2 Lead V4 Lead V6
<u>>ChannelSensitivity</u>	(003A,0210)	Ritmo: StripData
<u>>ChannelSensitivityUnits Sequence</u>	(003A,0211)	
<u>1 ITEM</u>		
>>CodeValue	(0008,0100)	"uV"
>>CodeSchemeDesignator	(0008,0102)	"UCUM"
>>CodeSchemeVersion	(0008,0103)	"1.4"
>>CodeMeaning	(0008,0104)	"microvolt"
<u>>ChannelSensitivityCorrectionFactor</u>	(003A,0212)	"1.0"
<u>>ChannelBaseline</u>	(003A,0213)	"0.0"
<u>>ChannelSampleSkew</u>	(003A,0215)	"0.0"

>WaveformBitsStored	(003A,021A)	"16"
>FilterLowFrequency	(003A,0220)	Filter Setting HightPass
>FilterHightFrequency	(003A,0221)	Filter Setting LowPass
>NotchFilterFrequency	(003A,0222)	Filter Setting Filter 50 Hz Filter 60 Hz
<u>WaveformBitsAllocated</u>	(5400,1004)	"16"
<u>WaveformSampleInterpretation</u>	(5400,1006)	"SS"
<u>waveformPaddingValue</u>	(5400,100A)	"00/ff"
<u>WaveformData</u>	(5400,1010)	Ritmo: <StripData>

Conclusions

Com ha conclusió d'aquest projecte de digitalització d'imatge no radiològica i en concret de la digitalització de les proves d'esforç podem afirmar que s'ha arribat a complir els objectius plantejats inicialment, s'ha aconseguit desenvolupar una aplicació que passi d'un format xml a un format dcm de manera automàtica.

Evidentment no han sigut els únics, per l'elaboració d'un projecte final de carrera es necessari portar a terme un ampli procés d'estudi, anàlisi i disseny de la idea que es vol desenvolupar. D'aquesta manera després de la proposta de la creació d'una aplicació per passar les dades a un format digital, es va realitzar una cerca d'aplicacions realitzades, així com un llistat de les necessitats més importants.

Acabada la primera fase de recollida dels requisits es va passar a pensar un disseny que tingués com a característica més important la facilitat d'incorporació de nous fabricant. Un cop pensada l'aplicació es va fer un primer disseny de classes i de seqüències; un cop van veure que el disseny era correcte es va passar a fer un disseny més acurat i es van definir fases de treball; les primeres fases consistien en construir l'aplicació i l'última fase va ser la incorporació d'un fabricant, que en el nostre cas va ser General Electrics amb l'aparell Mac1200.

Durant la realització del projecte ens vam adonar que hi havia altre disseny d'estructures que facilitaria la incorporació dels fabricants a l'aplicació (en l'apartat de viabilitat d'integració mitjançant Xslt ho detallo) però pensant en el temps de finalització del projecte no es va poder portar a terme la idea.

La realització d'aquest projecte m'ha permès adquirir coneixements en el camp de la interoperabilitat mèdica amb l'estudi de l'estàndard Dicom (he après com treballar amb ell i aplicar-lo d'una manera correcta), també he après quines són les metodologies a seguir per una bona gestió de projectes.

Bibliografia

- Herman Oosterwijk, *Dicom Bàsico (Segunda Edición)*. OTech Inc 2002
- Oleg S.Pianykh, *Digital Imaging and Communications in Medicine*. Springer 2008
- ARENAS LEON JM. *Electrocardiografía Normal y Patológica. 2*. McGraw-Hill 2000.
- Robert C.Martin, *UML for Java Programmers*. Prentice Hall 2003
- HOLZNER, STEVEN, *La Biblia de Java2*. Anaya 2000
- GE, *XML Data Export Reference Manual*
- <http://medical.nema.org> (on estan els estàndards dicom)
- <http://www.leadtools.com/sdk/medical/dicom.htm>
- <http://www.cabiatl.com/mricro>
- http://idlastro.gsfc.nasa.gov/idl_html_help/DICOM_Attributes.htm
- <http://www.dcm4che.org/> (llibreries dicom)
- http://www.otechimg.com/pdf/dicombasicsspanish_ch1.pdf

