INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA

ESCUELA DE INGENIERÍA Y TECNOLOGÍA

DESARROLLO DE UNA INTERFAZ DICOM PARA LA INTEGRACIÓN DE UNA FAMILIA DE MONITORES DE MECÁNICA RESPIRATORIA A UNA HISTORIA CLÍNICA ELECTRONICA

AUTOR: Gostuski, Vladimir Rista (Leg. Nº 51314)

PROFESOR RESPONSABLE: Madorno, Matias

TRABAJO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO DE BIOINGENIERO

BUENOS AIRES

PRIMER CUATRIMESTRE, 2017

Contenido

Introducción 4

Objetivo 6

Monitores de mecánica respiratoria 7

Digital Images and Communications in Medicine (DICOM) 9

Introducción 9

Modelo de Información DICOM 10

Flujo de trabajo en DICOM 11

Codificación de la información en DICOM 12

Definición de Objeto de Información (IOD, *Information Object Definition*) [6][7] 12

Elemento de Datos (DE, *Data Element*) [6][8] 15

Etiqueta (ET,*Tag*) 15

Valor de Representación (VR, *Value Representation*) 16

Largo del Valor (LV, *Value Length*) y Campo de Valor (CV, *Value Field*) 16

Codificación de Elementos de Datos 19

VR de Secuencia (SQ, *Sequence VR*) 20

Identificador Único (UID, *Unique Identifier*) 21

Diccionario de Datos DICOM (DDD, *DICOM Data Dictionary*) [9] 22

Estructura de un archivo DICOM[6][10] 23

Declaración de Conformidad DICOM (*DICOM Conformance Statement*) 24

Formas de Onda en DICOM [7] 25

Codificación de formas de onda 26

Comunicaciones en DICOM[6] 28

Materiales y herramientas 32

Materiales 32

Herramientas 32

Grassroots DICOM (GDCM) 32

CMake 32

Visualizadores DICOM 33

FluxView Flux Review 33

Matlab 33

Archivo modelo de ECG DICOM 34

Emuladores de SCU y SCP de Almacenamiento 34

Desarrollo de interfaz DICOM 35

Armado de estructuras DICOM con GDCM 35

Archivo y Set de Datos 35

Elementos de Datos 36

UIDs 36

Secuencias 36

Diagrama de flujo 39

Resultados 40

Generación de archivos DICOM 40

Ensayos con emuladores del Storage SOP 44

Pasos a seguir 46

Conclusiones 47

Bibliografía 48

Introducción

Los monitores FluxMed son una familia de equipos utilizados para adquirir señales y medir parámetros de la mecánica respiratoria en pacientes con respiración comprometida. Por pedido de algunos clientes surgió la necesidad de que estos monitores se puedan integrar a una historia clínica electrónica (HCE), en inglés, *hospital information system* (HIS). Como una opción para integrar a estos monitores a una HCE se eligió trabajar con el estándar Digital Images and Communications in Medicine (DICOM).

El Institute of Medicine (IOM) define a la HCE como *"aquella que reside en un sistema electrónico específicamente diseñado para recolectar, almacenar, manipular y dar soporte a los usuarios en cuanto a proveer accesibilidad a datos seguros y completos, alertas, recordatorios y sistemas clínicos de soporte para la toma de decisiones, brindando información clínica importante para el cuidado de los pacientes"* [1]. Frente a la historia clínica física (papel), las HCE mejoran la legibilidad, la accesibilidad y disponibilidad de la información y el almacenamiento, evitan la información por duplicado, no deterioran la información y permiten la integración de la información, entre otras mejoras [2].

Un requisito necesario de las HCEs es la interoperabilidad, definida en general por el Institute of Electrical and Electronics Engineers (IEEE) como la *"La capacidad de dos o más sistemas o componentes para intercambiar información y hacer uso de ella"* [3]. En el marco de la salud, la Healthcare Information and Management Systems Society (HIMSS) define a la interoperabilidad como *"la capacidad de diferentes sistemas de información en salud (sistemas hospitalarios, departamentales, registros clínicos electrónicos, etc.) para intercambiar datos y usar la información que ha sido intercambiada dentro y a través de los límites de la organización, con el fin de mejorar la prestación efectiva de los cuidados de salud a individuos y comunidades"* [4].

La interoperabilidad tiene un componente sintáctico y un componente semántico. La interoperabilidad sintáctica es el intercambio de datos, mensajes, documentos, según una estructura y sintaxis bien definida. El componente semántico implica que haya un entendimiento a nivel de interoperabilidad que se asocia con el significado de los contenidos y se refiere a la interpretación humana del contenido más que de la máquina. Por lo tanto, interoperabilidad en este nivel significa que hay un entendimiento común entre personas sobre el significado del contenido (información) que se intercambia (se garantiza la correcta interpretación y uso de la información intercambiada, para lo cual se necesitan definiciones
formales de cada entidad, atributo, relación, restricción y termino intercambiado). [5]

Para que dos o más sistemas sean interoperables se pueden crear interfaces entre ellos, pero esto es un problema cuando se quiere trabajar con una gran cantidad de sistemas. Los estándares facilitan la interoperabilidad al permitir comunicación entre aplicaciones con independencia del fabricante, la tecnología, o el lenguaje de desarrollo.

Considerando que las HCEs son cada vez más comunes en las instituciones de la salud, todo sistema aplicado a la salud debería considerar tener la capacidad de ser integrable con ellas. Esto incluye al equipamiento médico, siendo un caso emblema el del desarrollo de DICOM para lograr interoperabilidad entre sistemas relacionados al diagnóstico por imágenes. Algunos ejemplos de equipos de diagnóstico por imágenes son equipos de rayos X, tomografía computada, resonancia magnética, ultrasonido, medicina nuclear, etc. Con la llegada de la tecnología digital, estos equipos comenzaron a conectarse con impresoras, estaciones de trabajo y unidades de almacenamiento. La gran variedad de dispositivos disponibles y de protocolos de comunicación privativos imposibilitaban conectar entre sí equipos de distintos fabricantes. DICOM es un estándar ISO (12052:2017) y es el estándar de referencia para diagnóstico por imágenes a nivel global. Es prácticamente imposible manejar un centro de diagnóstico por imágenes sin implementarlo, ya que brinda interoperabilidad entre equipos de adquisición, estaciones de trabajo, impresoras y el *Picture Archiving and Communications System* (PACS), donde se almacenan y se solicitan las imágenes para ser utilizadas.

DICOM también define formatos para formas de onda, hemodinámicas, respiratorias, de pulso arterial y para audio y distintas modalidades de electrocardiograma. Se suelen usar en conjunto con estudios por imágenes, aunque también por sí solos y existen equipos que exportan este tipo de mediciones en formato DICOM.

Otro estándar en medicina digital es HL7, diseñado para el intercambio de mensajes clínicos y administrativos y desarrollado por la organización Health Level 7 International. El intercambio de información en las HCE se da por medio de HL7 y la integración de los PACS a las HCE por medio de interfaces DICOM-HL7. Tal es el caso de BetterCare, un sistema de registro de señales médicas en formato DICOM integrable a HCEs por medio de HL7. Este sistema desarrollado en Barcelona, al cual eventualmente se integrarán los monitores FluxMed, está creciendo como sistema de registro para señales respiratorias.

Para el desarrollo de integración realizado en este trabajo fue fundamental familiarizarse con el estándar DICOM y las herramientas para trabajar con él. Por lo tanto, una gran parte del informe está dedicado a explicar los componentes más básicos de DICOM, y está redactado de forma que un alumno avanzado de bioingeniería pueda introducirse en el tema.

Objetivo

La interoperabilidad sintáctica en DICOM tiene dos componentes: el formato que debe llevar la información (imágenes, formas de onda, etc.) y la comunicación entre aplicaciones. El alcance de este proyecto está limitado a desarrollar el primer componente, tomando los archivos generados por los equipos FluxMed y convirtiéndolos a formato DICOM.

Desarrollar esta herramienta como una aplicación que se ejecuta desde la línea de comando permite integrarla a otros software que ejecuten este proceso, dándole versatilidad en la integración con otros dispositivos.

Monitores de mecánica respiratoria

La familia de equipos FluxMed son monitores de mecánica respiratoria con la capacidad de medir desde tres hasta ocho señales, según el modelo. Además de la medición de estas señales también calculan parámetros de la mecánica respiratoria. La versión más completa puede medir señales de volumen y flujo de la respiración, presión en la vía aérea, presión esofágica, presión gástrica, presión transpulmonar, presión diafragmática y niveles de dióxido de carbono (CO_2) en la vía aérea (capnografía).

Las señales de flujo, presión esofágica, presión gástrica y capnografía son medidas en el paciente con una frecuencia de muestreo de 256Hz, y el resto de las señales son calculadas a partir de éstas. El almacenamiento de señales no tiene un límite de tiempo, aunque el uso habitual rara vez supera una o dos horas.

Todas las señales, excepto la de volumen, tienen un dígito significativo después del punto decimal, y el rango de medición varía entre señales. La señal de flujo puede tomar valores entre ±180L/min, las de presión ±75L/min o ±200cmH_2O, y las de capnografía de 0mmHg a 150mmHg.

Los equipos se pueden conectar vía USB a una PC para visualizar las señales en tiempo real con el software FluxView y también para almacenar las mediciones en un archivo de texto. El Cuadro 1 muestra el formato de los archivos de texto generados, que comienzan con un encabezado y luego siguen con las señales por columnas. El software FluxReview permite abrir los archivos de señales para visualizar dichas señales y los otros parámetros calculados como se muestra en la Figura 1. Todos los parámetros se calculan a partir de las señales medidas, con lo que estas señales originales son la información vital para almacenar.

FluxView - FluxMed acquisition system

Signal file Serial Number : 1.20h-

SW version: OX-CO2 File version: 1.13

Sampling rate: 256 Hz

Time	Flow	Volume	Paw	Pes	Ptpulm	Pga	Ptdiaf	CO2
0.00390625	0.0	0	0.0	23.9	-23.9	0.0	-23.9	0.0
0.0078125	0.0	0	0.0	23.9	-23.9	0.0	-23.9	0.0
0.01171875	0.0	0	0.0	23.9	-23.9	0.0	-23.9	0.0
0.015625	0.0	0	-0.1	23.8	-23.9	0.0	-23.8	0.0
0.01953125	0.0	0	-0.1	23.8	-23.9	0.0	-23.8	0.0
0.0234375	0.0	0	0.0	23.9	-23.9	0.0	-23.9	0.0
0.02734375	0.0	0	0.0	23.8	-23.8	0.0	-23.8	0.0
0.03125	0.0	0	-0.1	23.8	-23.9	0.0	-23.8	0.0
0.03515625	0.0	0	0.0	23.8	-23.8	0.0	-23.8	0.0
0.0390625	0.0	0	-0.1	23.6	-23.7	0.0	-23.6	0.0
0.04296875	0.0	0	0.0	23.6	-23.6	0.0	-23.6	0.0
0.046875	0.0	0	0.0	23.8	-23.8	0.0	-23.8	0.0

Cuadro 1. Formato de archivo de texto generado por los equipos FluxMed.

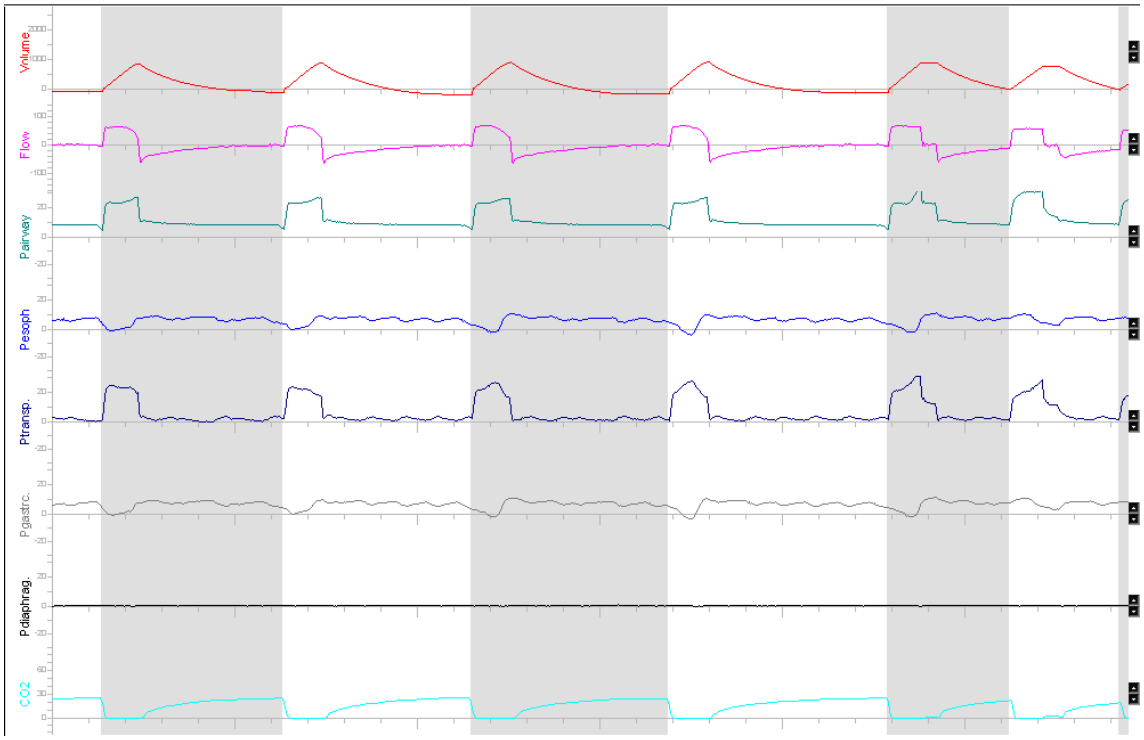


Figura 1. Visualización de señales en FluxReview.

Digital Images and Communications in Medicine (DICOM)

Introducción

DICOM es el estándar más difundido para el almacenamiento y transferencia de imágenes médicas. Es un estándar muy amplio que abarca formatos de archivo, protocolos de comunicación, almacenamiento y visualización y, por lo tanto, a veces es considerado un conjunto de estándares. Fue desarrollado por el *North American Manufacturers Association* (NEMA) en conjunto con el *American College of Radiology* (ACR) para brindar interoperabilidad entre equipos de distintos fabricantes.

La primera versión, llamada ACR/NEMA 300 fue publicada en 1985 y la segunda, ACR/NEMA V2.0, en 1988 obtuvo mayor aceptación y adhesión por parte de fabricantes. La tercera versión pasó a llamarse DICOM 3.0, fue publicada en 1993 y se actualiza con revisiones anuales. La nomenclatura de cada revisión está dada por el número de parte del estándar y el año de la revisión según el formato PS3.X-YYYY. La última versión de DICOM 3.0 consta de 21 partes y fue publicada en 2017. Entonces, por ejemplo, la última versión de la parte cinco se llama PS3.5-2017. El estándar completo, solamente disponible en inglés, se puede leer y descargar gratuitamente desde la página oficial de DICOM[1].

[1] https://www.dicomstandard.org/current/.

Modelo de Información DICOM

El Modelo de Información DICOM (Figura 2) es una abstracción de lo que ocurre cuando un paciente visita un centro de salud y se realiza un estudio. Es un modelo con una estructura jerárquica que empieza con el paciente, el cual debe tener una identificación (ID) dentro del centro de salud que lo identifique individualmente y de forma inequívoca. El paciente puede realizarse uno o varios estudios dentro del centro de salud (tomografía computada, resonancia magnética, ecografía, electrocardiograma), los cuales pueden contener una o múltiples series. Las series son conjuntos de una o más instancias (imágenes, formas de onda, reportes estructurados) generadas por un mismo equipo.

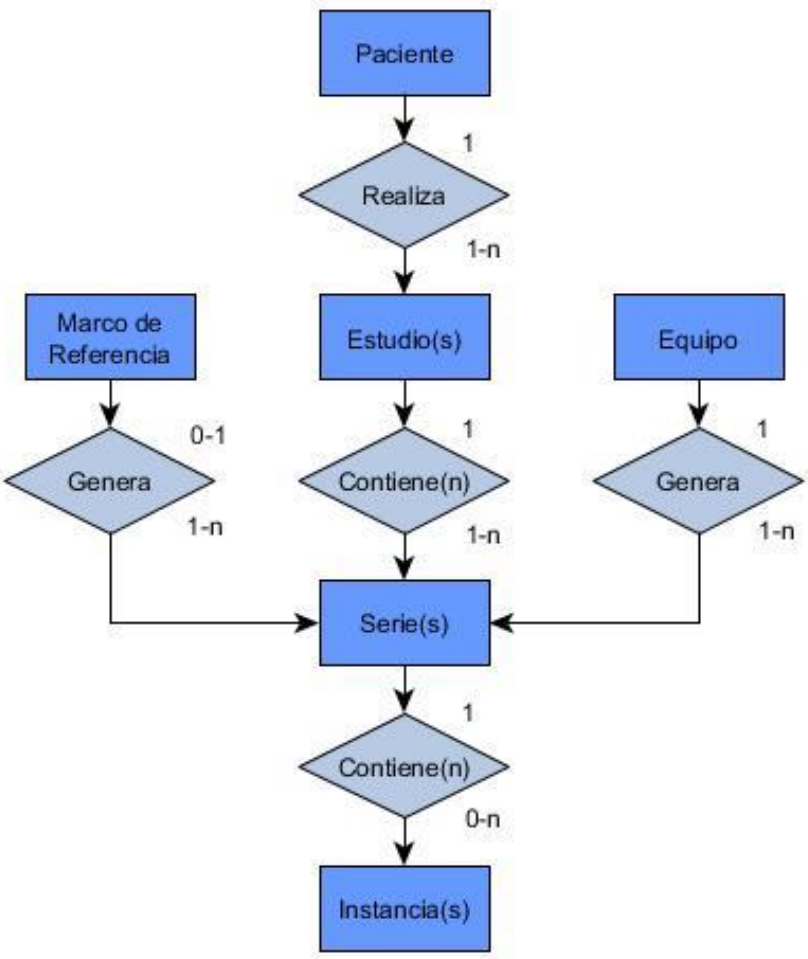


Figura 2. Modelo de Información DICOM

Flujo de trabajo en DICOM

Los equipos médicos, llamados modalidades en DICOM, adquieren instancias y generan archivos que las contienen. Estos archivos envían y son almacenados en el PACS. Desde ahí son solicitados desde las estaciones de trabajo, también llamadas *workstations*, para ser visualizados, editados, generar reportes a partir de ellos, etc. Esta dinámica de trabajo se puede ver en la Figura 3.

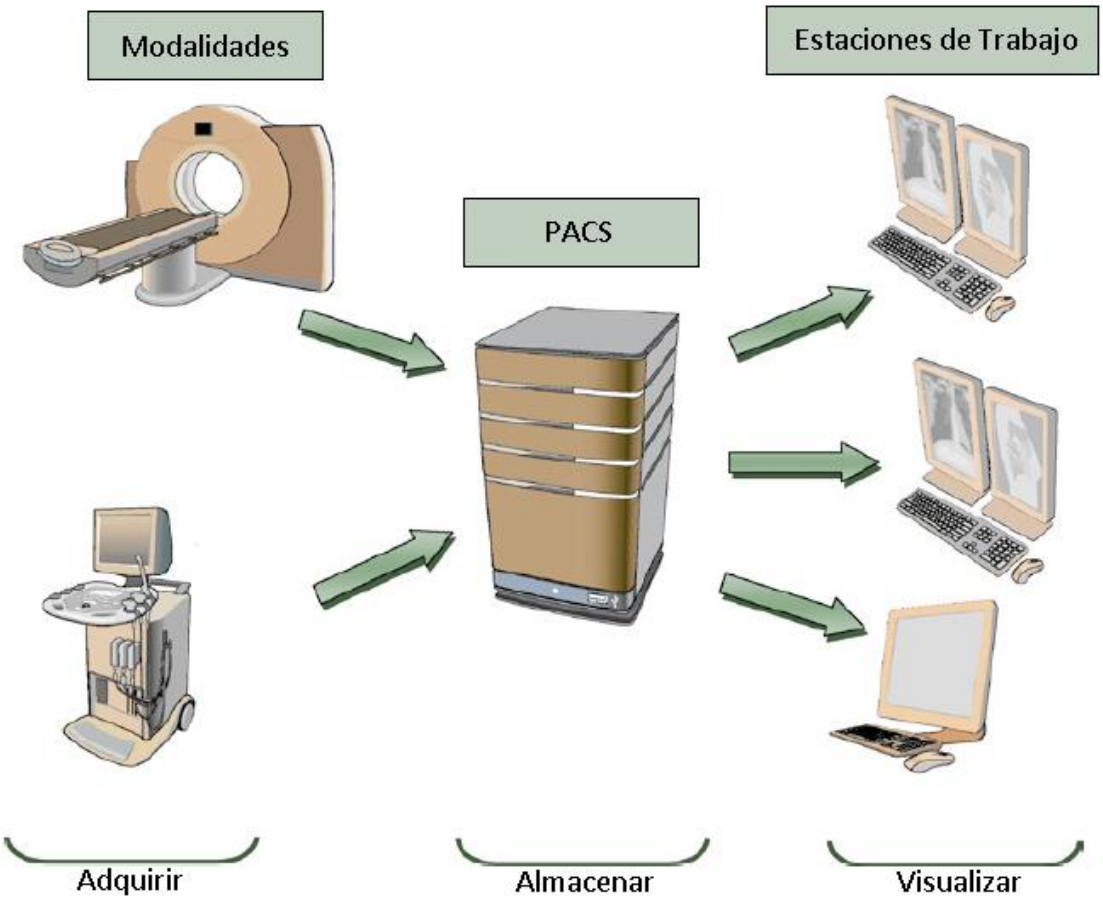


Figura 3. Flujo de trabajo en DICOM ([6], Fig 1.1).

Codificación de la información en DICOM

Definición de Objeto de Información (IOD, *Information Object Definition*) [6] [7]

Las IODs son conjuntos de datos que representan al Modelo de Información DICOM. Cada tipo de instancia en DICOM tiene una IOD asociada, como por ejemplo la IOD de Imagen Tomografía Computada (Tabla 1), la IOD de Imagen Medicina Nuclear, la IOD de Reporte Estructurado de Texto Básico, la IOD de Forma de Onda de ECG General. Las IODs están organizadas en bloques de información con una estructura jerárquica, cuyo primer nivel está construido a partir de la Entidad de Información (IE, *Information Entity*). Como se puede ver en la Tabla 1, las IEs representan a los mismos bloques definidos en el Modelo de Información DICOM. Todas las IODs están conformados por las mismas IEs, que a su vez se construyen a partir del Módulo de Información (IM, *Information Module*). El uso de un IM dentro de una IOD puede ser mandatorio (*mandatory*), representado por una "M", o a elección del usuario (*user-defined*), representado por una "U". Los IM a su vez están subdivididos en el bloque más básico, Elemento de Datos (DE, *Data Elements*). La Tabla 2 algunos de los atributos que conforman el IM Paciente (*Patient)*. Cuando los DEs de una IOD toman valores se genera un Set de Datos (DS, *Data Set*).

Tabla 1. IOD de Tomografía Computada.

EI	Modulo	Referencia	Uso
Paciente	Paciente	C.7.1.1	M
	Sujeto de Prueba Clínica	C.7.1.3	U
Estudio	Estudio General	C.7.2.1	M
	Estudio del Paciente	C.7.2.2	U
	Estudio de Prueba Clínica	C.7.2.3	U
Serie	Serie General	C.7.3.1	M
	Serie de Prueba Clínica	C.7.3.2	U
Marco de Referencia	Marco de Referencia	C.7.4.1	M
Equipamiento	Equipamiento General	C.7.5.1	M
Imagen	Imagen General	C.7.6.1	M
	Plano de imagen	C.7.6.2	M
	Píxel de imagen	C.7.6.3	M
	Contraste/Bolo	C.7.6.4	C – Requerido si se usó contraste en la imagen
	Dispositivo	C.7.6.12	U
	Espécimen	C.7.6.22	U
	Imagen de TC	C.8.2.1	M
	Plano de Superposición	C.9.2	U
	SOP Común	C.12.1	M
	Referencia de Instancia Común	C.12.2	U

Tabla 2. Atributos del IM Paciente.

Nombre del atributo	Etiqueta	Tipo	Descripción del Atributo
Nombre del Paciente	(0010,0010)	2	Nombre completo del paciente.
ID del Paciente.	(0010,0020)	2	Numero o código de identificación principal del paciente.
Fecha de Nacimiento del Paciente	(0010,0030)	2	Fecha de Nacimiento del paciente.
Sexo del Paciente.	(0010,0040)	2	Sexo del paciente nombrado. Valores enumerados: M masculino F femenino O otro
Secuencia de Foto del Paciente Referenciado	(0010,1100)	3	Una fotografía que confirma la identidad del paciente. Solo se permite incluir un Ítem en esta Secuencia. Ver C.2.2.1.1.
Secuencia del Paciente Referenciado	(0008,1120)	3	Una secuencia que provee referencia a un SOP de par Clase/Instancia del Paciente. Solo se permite incluir un Ítem en esta Secuencia.
Hora de Nacimiento del Paciente	(0010,0032)	3	Hora de Nacimiento del Paciente.
Otros IDs del Paciente.	(0010,1000)	3	Otros números o códigos de identificación usados para identificar al paciente.
Secuencia de Otros IDs del Paciente	(0010,1002)	3	Una secuencia de números o códigos usados para identificar al paciente, que pueden o no ser legibles por humanos, y pueden o no haber sido obtenidos de un dispositivo implantado o sujetado tal como un ID de radiofrecuencia o un código de barras. Esta secuencia permite incluir uno o más Ítems.

Elemento de Datos (DE, *Data Element*) [6][8]

Los DEs está conformados por una Etiqueta (ET, *Tag*), un Valor de Representación (VR, *Value Representation*), un Largo del Valor (VL, *Value Length*) y un Campo de Valor (CV, *Value Field*) que contiene el valor correspondiente al atributo según la instancia. La Figura 4 muestra los componentes de un DE por orden de aparición en su estructura interna.

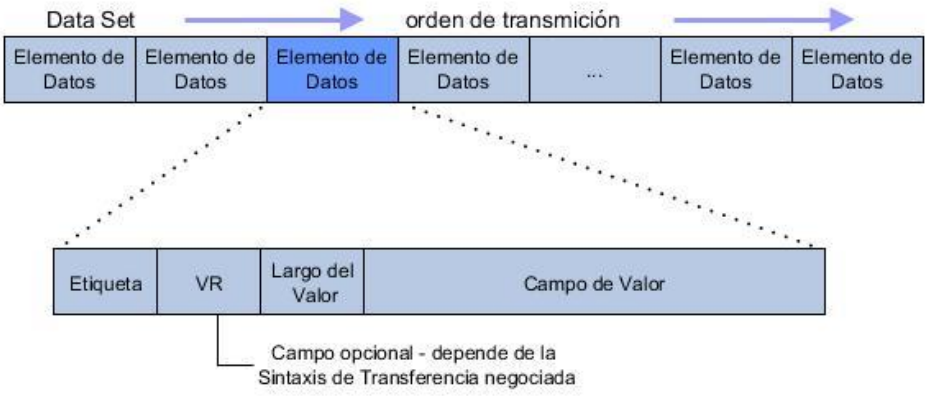


Figura 4. Componentes de un DE o atributo.

Etiqueta (ET, *Tag*)

Las ETs son un par de números representados en hexadecimal que identifican individualmente a cada DE, según la estructura (*número de grupo, número de elemento)*. Los atributos están agrupados según lo que definen, por ejemplo, el número de grupo "10" está conformado por todos los atributos relacionados con la información general del paciente. El número de elemento dentro de un grupo se asigna únicamente a un atributo y de esta manera se lo identifica. El largo total de una ET son cuatro bytes (dos bytes para el número de grupo y dos bytes para el número de elemento).

Valor de Representación (VR, *Value Representation*)

Los VRs son especificaciones de tipo de dato y de formato para los valores de los DEs. Hay 27 VRs definidos y toda la información escrita en DICOM tiene que coincidir con alguno de ellos. Los VRs tienen un nombre que los describe con una abreviatura de dos letras, una descripción, un repertorio de caracteres válidos y una longitud válida para el valor del DE. Los VRs pueden definir DEs en formato de texto, nombres, fechas, IDs, etc (Tabla 3); o en binario; valores de pixeles o muestras de una curva (Tabla 4).

Largo del Valor (LV, *Value Length*) y Campo de Valor (CV, *Value Field*)

En DICOM el valor de cada DE se registra junto a su tamaño, que puede ser fijo o limitado según el VR correspondiente. El largo total de un valor se especifica en el campo LV, en un espacio de cuatro o dos bytes y el valor del atributo en el CV, que debe tener el largo especificado por el campo LV. Una característica del largo de los valores en DICOM ya sea en caracteres o bytes, es que siempre deben ser pares. Los valores impares tienen que ser rellenados con un espacio en blanco.

Tabla 3. Algunos ejemplos de VR con formato de texto.

Nombre del VR	Definición	Repertorio de Caracteres	Largo del Valor
AE Entidad de Aplicación	Una cadena de caracteres que identifica a una Entidad de Aplicación, donde los espacios (20H) posteriores y previos son no-significativos. No se puede utilizar un valor que solamente consista de espacios.	Repertorio de Caracteres Default excluyendo el código de carácter 5CH (la BARRA INVERSA "\" en ISO-IR 6), y todos los caracteres de control.	16 bytes máximo
AS Cadena de Edad	Una cadena de caracteres con uno de los siguientes formatos—nnnD, nnnW, nnnM, nnnY; donde nnn debe contener el número de días para D, semanas para W, meses para M, o años para Y. Ejemplo: "018M" representa una edad de 18 meses.	"0"-"9", "D", "W", "M", "Y" del Repertorio de Caracteres Default	4 bytes fijos
AT Etiqueta del Atributo	Para ordenado de enteros sin signar de 16 bits que es el valor de una Etiqueta de un Elemento de Datos. Ejemplo: Una Etiqueta de un Elemento de Datos con valor (0018,00FF) se codificaría en una serie de 4 bytes en una Sintaxis de Transferencia con orden en Little-Endian como 18H,00H,FFH,00H La codificación de un valor de AT es exactamente igual a la codificación de una Etiqueta de Elemento de Datos como se define en la Sección 7.	No aplica	4 bytes fijos
CS Cadena de Código	Una cadena de caracteres que identifica un concepto controlado. Los espacios (20H) previos y posteriores no son significativos.	Caracteres en mayuscula, "0"-"9", el carácter ESPACIO y guion bajo "_" del Repertorio de Caracteres Default.	16 bytes máximo

Tabla 4 Algunos ejemplos de VR en formato binario.

VR Name	Definition	Character Repertoire	Length of Value
FL Punto Flotante Simple	Número binario de punto flotante con precisión simple representado en formato IEEE 754:1985 Formato de Número de Punto Flotante de 32-bits.	No aplica	4 bytes fijos
FD Punto Flotante Doble	Número binario de punto flotante de precisión doble representado en formato IEEE 754:1985 Formato de Número de Punto Flotante de 64-bits.	No aplica	8 bytes fijos
IS Cadena de Enteros	Una cadena de caracteres que representa un entero en base-10 (decimal), deberá contener solamente los caracteres 0 – 9, con un "+" o "-"previo opcional. Puede ser rellenado con espacios tanto al principio como al final. Los espacios embebidos no están permitidos. El entero n representado deber estar en el rango: $-2^{31} <= n <= (2^{31}-1)$.	"0"-"9", "+", "-" y el carácter ESPACIO del Repertorio de Caracteres Default.	12 bytes máximo
LO Cadena Larga	Una cadena de caracteres que puede ser rellenada con espacios tanto al principio como al final. El código de caracter 5CH (la BARRA INVERSA "\" en ISO-IR 6) no debe estar presente, ya que es utilizada como el delimitador entre valores en elementos de datos con valores múltiples. La cadena no debe tener Caracteres de Control excepto por ESC.	Repertorio de Caracteres Default y/o como se define en (0008,0005), excluyendo el código de carácter 5CH (la BARRA INVERSA "\" en ISO-IR 6), y todos los caracteres de control excepto por ESC cuando es usado para secuencias de escape ISO 2022.	64 chars máximo (ver Nota en la Sección 6.2)
LT Texto Largo	Una cadena de caracteres que puede contener uno o mas párrafos. Puede contener el set de Caracteres Gráficos y los Caracteres de Control CR, LF, FF y ESC. Puede ser rellenado con espacios al final, los cuales pueden ser ignorados pero los espacios al principio se consideran significativos. Los Elementos de Datos con este VR no deberán tener valores múltiples y por lo tanto el código de carácter 5CH 5CH (la BARRA INVERSA "\" en ISO-IR 6) puede ser usada.	Repertorio de Caracteres Default y/o como se define en (0008,0005), excluyendo Caracteres de Control excepto TAB, LF, FF, CR (y ESC cuando es usado para secuencias de escape ISO 2022).	10240 chars máximo (ver Nota en la Sección 6.2)
OB Otro Byte	Un stream de octetos donde la codificación de los contenidos es especificada por la Sintaxis de Transferencia negociada. OB es un VR que es sensible al orden de los bytes (ver Sección 7.3). El stream de octetos debe ser rellenado con un único byte de valor NULL (00H) al final cuando sea necesario para llegar a un largo par.	No aplica	Ver definición de la Sintaxis de Transferencia

Codificación de Elementos de Datos

Hay dos clases de codificaciones para DEs, Codificación de VR Implícita (*Implicit VR encoding*) (Tabla 5), que es la codificación por default, y Codificación de VR Explícita (*Explicit VR encoding*), que tiene dos subdivisiones según el tipo de VR que se use (Tabla 6 y Tabla 7). La diferencia principal entre estos tipos de codificación es que la explícita reserva dos o cuatro bytes para especificar el tipo de VR que se está usando. Al armar un DS, los DEs son concatenados por orden de número de grupo creciente y luego dentro de cada grupo por número de elemento creciente. El orden de los bytes por default en DICOM es *Little Endian*, es decir, los datos se ordenan con el byte menos significativo primero y el más significativo último.

Tabla 5. Codificación de VR Implícita.

Etiqueta		Largo del Valor	Valor
Número de Grupo (Entero sin signar de 16 bits)	Número de Elemento (Entero sin signar de 16 bits)	Entero sin signar de 32 bits	Número de bytes par que contiene el Valor del Elemento de Datos codificado según el VR especificado en PS3.6 y la Sintaxis de Transferencia negociada. Si es de Largo Indefinido se delimita con el Ítem de Delimitación de Secuencia
2 bytes	2 bytes	4 bytes	Bytes de 'Largo del Valor' o Largo Indefinido

Tabla 6. Codificación de VR explícita para los VR de tipo OB, OD, OF, OL, OW, SQ, UC, UR, UT y UN.

Etiqueta		VR		Largo del Valor	Valor
Número de Grupo (Entero sin signar de 16 bits)	Número de Elemento (Entero sin signar de 16 bits)	VR (2 caracteres de un byte) del tipo "OB", "OD", "OF", "OL", "OW", "SQ", "UC", "UR", "UT" or "UN"	Reservado (2 bytes) con un valor de 0000H	Entero sin signar de 32 bits	Numero de bytes para que contiene el/los Valor(es) del Elemento de Datos codificado según el VR y la Sintaxis de Transferencia negociada. Si es de Largo Indefinido se delimita con el Ítem de Delimitación de Secuencia
2 bytes	2 bytes	2 bytes	2 bytes	4 bytes	Bytes de 'Largo del Valor' si es de Largo Explicito

Tabla 7. Codificación de VR Explícita para los otros tipos de VR.

Etiqueta		VR	Largo del Valor	Valor
Número de Grupo (Entero sin signar de 16 bits)	Element Number (Entero sin signar de 16 bits)	VR (2 caracteres de un byte)	(Entero sin signar de 16 bits)	Numero de bytes para que contiene el/los Valor(es) del Elemento de Datos codificado según el VR y la Sintaxis de Transferencia negociada.
2 bytes	2 bytes	2 bytes	2 bytes	Bytes de 'Largo del Valor'

VR de Secuencia (SQ, *Sequence VR*)

DICOM permite agrupar `DEs` relacionados por medio de secuencias, que son `DEs` con `VR` del tipo `SQ`. Las secuencias contienen ítems, que a su vez contienen `DS` anidados. El número de ítems que puede contener cada secuencia se define según la `IOD` correspondiente y se ordenan en orden creciente con siendo el primero indexado en la posición uno. Los ítems también pueden contener otras secuencias, y de esta manera se forman estructuras ramificadas como muestra la Figura 5.

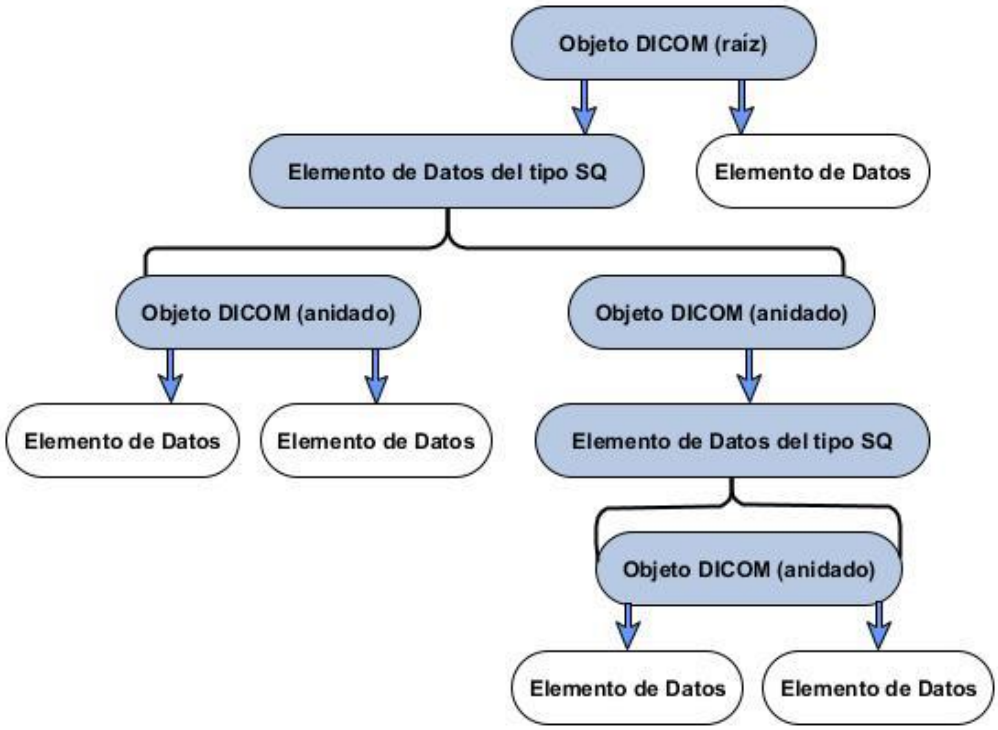


Figura 5. Estructura ramificada formada por el uso de secuencias anidadas([6], Fig 5.4).

Identificador Único (UID, *Unique Identifier*)

Los `UIDs` son el mecanismo que tiene DICOM para etiquetar instancias de objetos. Están generados de forma que no se repitan, con la intención de poder identificar estudios con unicidad dentro de una misma institución, país e inclusive entre distintos países. Son codificados según el `VR UI`, que es una cadena (*string*) de caracteres de números separados por puntos según la estructura <raíz org>.<sufijo> y con un largo máximo de 64 bytes. El primer parte es un ID raíz (*root ID)* que identifica a una organización, por ejemplo, a un fabricante de equipos de imágenes médicas. Las organizaciones que quieran trabajar con DICOM pueden solicitar a NEMA un ID raíz propia. La parte del sufijo (*suffix*) es la que garantiza la unicidad de los `UID`, puede, por ejemplo, seguir la estructura <ID del paciente>.<ID del estudio>.<fecha>.<tiempo en milisegundos>.

Los `UIDs` están relacionados con el Modelo de Información DICOM y cumplen un rol importante en la organización de la información dentro de los PACS. Cada paciente debe tener un ID propio dentro de un centro de salud, que se registrará en el atributo ID del Paciente (0010,0020) cada vez que ese paciente se realiza un estudio. Este dato puede llegar al equipo de forma automática o manual, dependiendo del equipo y del sistema de información del centro de salud. Los archivos dentro de un PACS se organizan de forma jerárquica por los `UIDs Paciente`, `Instancia de Estudio`, `Instancia de Serie` e `Instancia de SOP` en el orden mencionado (Figura 6). Esto quiere decir que los archivos que tienen un mismo `UID` en cualquiera de estos niveles serán agrupados. Por lo tanto, un error en el ingreso del ID del paciente puede resultar en un archivo perdido o en un archivo asignado a otro paciente. A diferencia del ID de Paciente, los otros `UID` son generados por los equipos automáticamente, minimizando la posibilidad de agrupar archivos erróneamente.

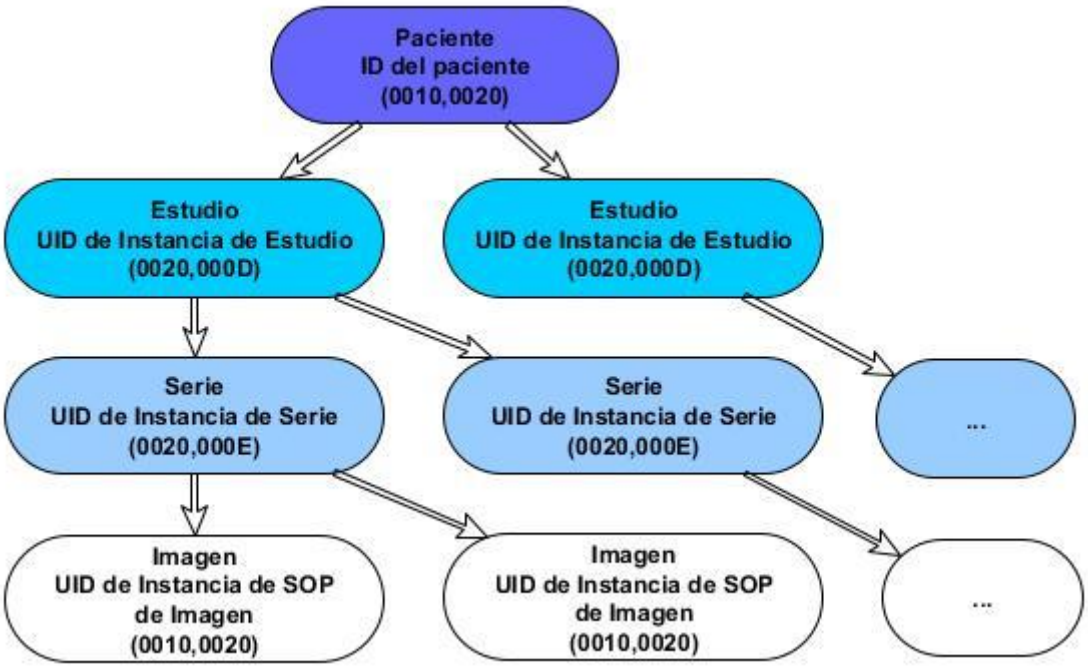


Figura 6. Organización jerárquica según UID.

Diccionario de Datos DICOM (DDD, *DICOM Data Dictionary*) [9]

El DDD es un listado de todos los atributos definidos en DICOM (son más de 2000 y hay nuevos con cada edición). Los atributos se encuentran ordenados en forma creciente de etiquetas, primero por número de grupo y luego por número de elemento, tal como se puede ver en la Tabla 8. Todos los números de grupo definidos son pares, los números impares están reservados para que los fabricantes puedan incluir atributos propios, denominados "privados".

El DDD también especifica una Palabra Clave (*Keyword*) para identificar cada atributo, el VR correspondiente para cada atributo y su Valor de Multiplicidad (VM, *Value Multiplicity*), que indica con cuantos valores pueden definir un atributo. Además de estar disponible en el estándar, hay varios sitios web que permiten realizar búsquedas del DDD[2].

Tabla 8. Parte del DICOM Data Dictionary.

Etiqueta	Nombre	Palabra Clave	VR	VM
(0008,0001)	*Largo Hasta el Final*	*LargoHastaFinal*	*UL*	*1*
(0008,0005)	Set de Caracteres Específicos	SetCaracteresEspecificos	CS	1-n
(0008,0006)	Secuencia de Código de Lenguaje	SecuenciaCodigoLenguaje	SQ	1
(0008,0008)	Tipo de Imagen	TipoImagen	CS	2-n
(0008,0010)	*Código de Reconocimiento*	*CodigoReconocimiento*	*SH*	*1*
(0008,0012)	Fecha de Creación de la Instancia	FechaCreacionInstancia	DA	1
(0008,0013)	Hora de Creación de la Instancia	HoraCreacionInstancia	TM	1
(0008,0014)	UID de Creador de Instancia	UIDCreadorInstancia	UI	1
(0008,0015)	Coerción de Instancia de FechaDía	CoercionInstanciaFechaDia	DT	1
(0008,0016)	UID de Clase de SOP	UIDClaseSOP	UI	1
(0008,0018)	UID de Instancia de SOP	UIDInstanciaSOP	UI	1
(0008,001A)	UID de Clase General de SOP Relacionada	UIDClaseGeneralSOPRelacionada	UI	1-n
(0008,001B)	UID de Clase Original de SOP Especializada	UIDClaseOriginalSOPEspecializada	UI	1
(0008,0020)	Fecha de Estudio	FechaEstudio	DA	1

[2]https://dicom.innolitics.com/ciod

http://dicomlookup.com/lookuppage.asp

http://www.dicom-solutions.com/dicompedia.shtm?lang=en

Estructura de un archivo DICOM[6][10]

Todos los archivos DICOM comienzan con un encabezado (*header*), al que le sigue el `DS`. El encabezado está conformado, en orden de aparición, por un preámbulo, el prefijo DICM y la Metainformación del Archivo (*File Meta Information*), como muestra la Figura 7. El preámbulo consta de 128 bytes, sin ninguna estructura o valor definido, por lo cual se puede rellenar con ceros. El prefijo consta de los caracteres DICM en los bytes 129 a 132. La metainformación consta de los atributos del grupo 0002, son diez en total y deben codificarse usando codificación de `VR` explícita. Entre los obligatorios el más importante es el atributo `UID de Sintaxis de Transferencia` ((0002, 0010), *Transfer Syntax* UID) que especifica la sintaxis de transferencia usada en el archivo (por ejemplo, `VR Implícito` en *little endian*, `VR Explícito` en *little endian*, etc.). El largo de la metainformación está dado por el atributo `Largo del Grupo` ((0002,0000), *Group Length*). Terminado el encabezado, se coloca el `DS` u objeto DICOM, que puede incluir o no el `DE` nulo (FFFC,FFFC) al final. Los archivos DICOM contienen atributos en formato de texto y en formato binario, por lo cual para visualizar su contenido es necesario usar un software dedicado.

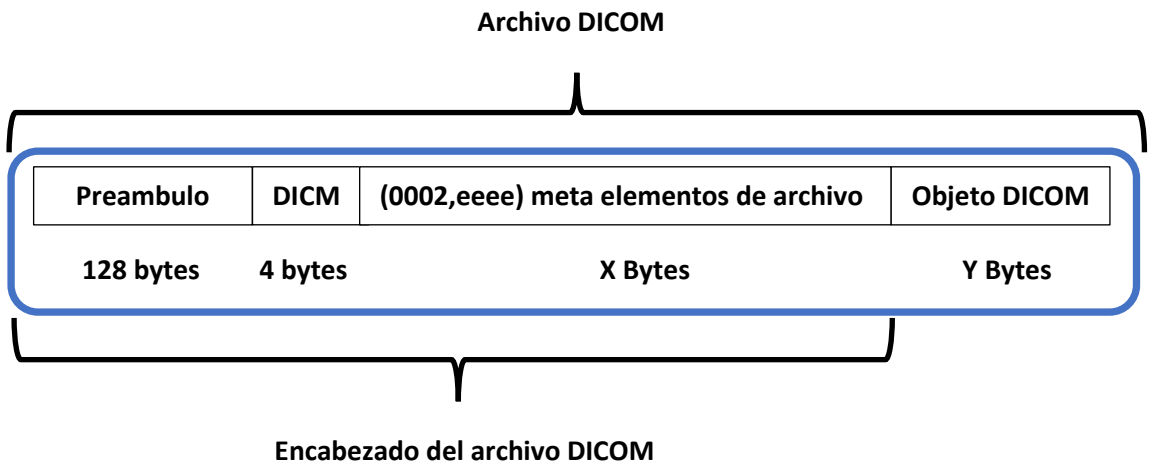


Figura 7. Estructura de un archivo DICOM ([5], Fig. 10.1).

Declaración de Conformidad DICOM (*DICOM Conformance Statement*)

Es una declaración formal de los productos DICOM para especificar cuan adherentes al estándar son las `IODs`, `SOPs` y protocolos de comunicación implementados en la aplicación. Es un documento de gran interés para los clientes ya que permite al momento de comprar un producto, evaluar si es compatible con el resto de sus aplicaciones DICOM. La confección de este documento está fuera del alcance de este trabajo, pero debería estar disponible para la instancia de comercialización.

La implementación de las `IODs` se detalla atributo por atributo en esta declaración. La Figura 8 muestra un segmento de una declaración de conformidad de los productos de ECG Quinton Q-Stress, Mortara XScribe, HSCribe, y RScribe de Mortara Instruments, una empresa de EEUU cuyos equipos tienen soporte DICOM para formas de onda de ECG.

Mortara INSTRUMENT

3.2.1.3.4.2 SOP Specific Conformance for 12-lead ECG Waveform Storage

Modality Manager creates 12-lead ECG Waveform objects with the following DICOM attributes. All attributes conform to the 12-lead ECG Waveform Storage SOP class specification. A private attribute block is included for the VERITAS interpretation vector, Attending Physician and Notes.

Table 3.2.1.3.4.2 12-lead ECG Waveform Storage Attributes

Module	Attribute Name	Tag	Value
Patient	Patient's Name	(0010,0010)	From MWL or Last Name, First Name and Middle Name fields
	Patient ID	(0010,0020)	From MWL or ID field
	Patient's Birth Date	(0010,0030)	From MWL or DOB field (yyyymmdd)
	Patient's Sex	(0010,0040)	From MWL or Gender field (M,F or O)
	Other Patient IDs	(0010,1000)	From MWL or Second ID field
	Ethnic Group	(0010,2160)	From MWL or Race field Caucasian, Black, Oriental, Hispanic, American Indian, Aleut/Eskimo, Hawaiian, Pacific Islander, Mongolian, Asian, Unknown. **Note:** This tag will not be present if the custom ID does not contain the Race field.
General Study	Study Instance UID	(0020,000D)	From MWL or generated by Modality Manager using the following : Mortara prefix: 1.3.6.1.4.1.20029 Product code for RScribe: 40 Exam entry date/time
	Study Date	(0008,0020)	Acquisition Date
	Study Time	(0008,0030)	Acquisition Time
	Referring Physician's Name	(0008,0090)	From MWL or Referring Physician field

Figura 8. Parte de la declaración de conformidad de Mortara Instruments.

Formas de Onda en DICOM [7]

DICOM ofrece nueve IODs para formas de onda Audio de Voz Básico (*Basic Voice Audio*), Electrocardiograma de 12 Derivaciones (*12-Lead Electrocardiogram*) (Tabla 9), Electrocardiograma General (*General Electrocardiogram*), Electrocardiograma Ambulatorio (*Ambulatory Electrocardiogram*), Forma de Onda Hemodinámica (*Hemodynamic Waveform*), Electrofisiología Cardíaca Básica (*Basic Cardiac Electrophysiology*), Forma de Onda de Pulso Arterial (*Arterial Pulse Waveform*), Forma de Onda Respiratoria (*Respiratory Waveform*) y Forma de Onda de Audio General (*General Audio Waveform*).

Todas estas IODs están compuestas por las mismas IEs, Paciente (*Patient*), Estudio (*Study*), Serie (*Series*), Marco de Referencia (*Frame of Reference*), Equipamiento (*Equipment*) y Forma de Onda (*Waveform*). Las diferencias entre ellas están en el requerimiento de uso para los IM (M o U) y en los valores que pueden tomar ciertos atributos. Por ejemplo, el Módulo de Sincronización (*Synchronization Module*) es opcional en la IOD ECG de 12 Derivaciones (*12-Lead ECG*) pero obligatorio en la IOD Forma de Onda Respiratoria (*Respiratory Waveform*) y el atributo Frecuencia de Muestreo (*Sampling Frequency*) puede tomar valores entre 200 y 1000 para la IOD ECG de 12 derivaciones, y solo valores menores a 100 para el IOD Forma de Onda Respiratoria.

Tabla 9. IOD ECG de 12 Derivaciones.

EI	Modulo	Referencia	Uso
Paciente	Paciente	C.7.1.1	M
	Sujeto de Prueba Clínica	C.7.1.3	U
Estudio	Estudio General	C.7.2.1	M
	Estudio del Paciente	C.7.2.2	U
	Estudio de Prueba Clínica	C.7.2.3	U
Serie	Serie General	C.7.3.1	M
	Serie de Prueba Clínica	C.7.3.2	U
Marco de Referencia	Sincronización	C.7.4.2	U
Equipamiento	Equipamiento General	C.7.5.1	M
Forma de Onda	Identificación de Forma de Onda	C.10.8	M
	Forma de Onda	C.10.9	M
	Contexto de Adquisición	C.7.6.14	M
	Anotación de Forma de Onda	C.10.10	C - Required if annotation is present
	SOP Común	C.12.1	M

Codificación de formas de onda

La mayoría de los atributos que caracterizan a las señales están en el IM `Forma de Onda` (*Waveform*). El atributo `Secuencia de Forma de Onda` (*Waveform Sequence*) es una secuencia de ítems denominados `Grupos de Forma de Onda Multiplexadas` (`WMG`, *Waveform Multiplex Groups*). En estos ítems se almacenan las señales, llamadas canales (*channels*) y los atributos que las describen. Los `WMG` se arman principalmente con los atributos listados en la Tabla 10.

Tabla 10. Principales atributos para armar un WMG.

Tag	Atributo	Descripción
(0x003a,0x0004)	Originalidad de Forma de Onda	Denota si las muestras son mediciones directas o si están derivadas de otras señales.
(0x003a,0x0005)	Número de Canales de Forma de Onda	Número de señales (canales o channels) por bloque de señales.
(0x003a,0x0010)	Número de Muestras de Forma de Onda	Número de muestras en cada canal.
(0x003a,0x001a)	Frecuencia de Muestreo	Frecuencia de muestreo de las señales.
(0x003a,0x0200)	Secuencia de Definición de Canal	Secuencia que define cada canal individualmente.
(0x5400,0x1004)	Bits de Forma de Onda Asignados	Número de bits que ocupa cada muestra.
(0x5400,0x1006)	Interpretación de Muestra de Forma de Onda	Como se debe interpretar cada muestra (8 bits signado, 8 bits sin signo, 16 bits signado, 16 bits sin signo).
(0x5400,0x1010)	Datos de Forma de Onda	Vector que contiene todas las muestras de un bloque de señales.

Cada `WMG` contiene un bloque de señales con un tamaño determinado por el número de canales, el número de muestras por canal y el formato de almacenamiento de las muestras. El número de canales se define en el atributo `Número de Canales de Forma de Onda` (*Number of Waveform Channels*). El número de muestras por canal, que es igual para todos los canales, se define en `Número de Muestras de Forma de Onda` (*Number of Waveform Samples*). El bloque de señales se guarda en el atributo `Datos de Forma de Onda` (*Waveform Data*) en un vector ordenado según la siguiente estructura y sin separación entre muestras,

$$C_1M_1, C_2M_1, C_3M_1, \dots, C_nM_1, C_1M_2, C_2M_2, C_3M_2, \dots, C_nM_2, \dots, C_nM_s$$

donde C_x denota el canal número *x* y *n* el número de canales definidos en `Número de Canales de Forma de Onda`. M_y denota la muestra número *y*, y *s* el número de muestras por canal definido en `Número de Muestras de Forma de Onda`.

Por lo tanto, el número total de muestras por `WMG` está dado por:

$$número\ de\ muestras\ por\ bloque\ de\ señales = C_n * M_s$$

El largo del bloque de señales en bytes se define con los atributos `Interpretación de Muestra de Forma de Onda` (*Waveform Sample Interpretation*) y `Bits de Forma de Onda Asignados` (*Waveform Bits Allocated*). Dependiendo de la `IOD` utilizada; cada muestra se puede representar con 8 o 16 bits (uno o dos bytes), como muestra la Tabla 11.

$$largo\ en\ bytes\ de\ cada\ bloque\ de\ señales = C_n * M_s * b, con\ b = 1{,}2$$

Finalmente, el largo en tiempo del bloque está dado por el número de muestras por canal y la frecuencia de muestreo f_s establecido en el atributo `Frecuencia de Muestreo` (*Sampling Frequency*).

$$largo\ en\ tiempo\ por\ bloque\ de\ señales = f_s * s$$

Tabla 11. Distintas formas de almacenar una muestra de forma de onda.

Bits de Forma de Onda Asignados – Terminos Definidos	Interpretación de Muestra de Forma de Onda – Terminos Definidos	Significado de Interpretación de Muestra de Forma de Onda
8	SB	Signado lineal de 8 bits
	UB	No signado lineal de 8 bits
	MB	Ley Mu de 8 bits (en concordancia con la Recomendación ITU/T G.711)
	AB	Ley A de 8 bits (en concordancia con la Recomendación ITU/T G.711)
16	SS	Signado lineal de 8 bits
	US	Signado lineal de 16 bits

$$largo\ en\ tiempo\ por\ bloque\ de\ señales = f_s * s$$

Comunicaciones en DICOM[6]

Las Entidades de Aplicación (AE, *Application Entities*) son equipos o programas que se encuentran dentro de una red DICOM. Las comunicaciones entre AE se dan por medio del Protocolo de control de transmisión (TCP, *Transmission Control Protocol*) e Protocolo de Internet (IP, *Internet Protocol*). Para que haya comunicación entre dos AEs hace falta definir:

1. (*AE Title*): es un nombre que se le designa individualmente a cada AE. Se codifica con el VR AE y deben ser preferentemente alfanuméricos.
2. Dirección IP de AE (*AE IP address*): es el número IP del dispositivo que contiene a la AE, puede ser compartido por más de una AE.
3. Número de puerto de AE (*AE port number*): es el número de puerto TCP. Por default el número de puerto para DICOM es 104 pero se le puede asignar cualquier valor desde 0 hasta 65535. El número de puerto debe ser asignado individualmente para cada AE ya que puede haber más de una AE funcionando en un mismo dispositivo, por ejemplo, una computadora con distintos software DICOM.

Las AE se demandan y proveen servicios entre sí. Cuando demandan un servicio se las denomina Usuarios de Clases de Servicio (SCU, *Service Class Users*) y cuando proveen un servicio se los denomina Proveedores de Clases de Servicios (SCP, *Service Class Providers*). El rol de SCU y SCP es relativo, ya que una misma AE puede ser SCU y SCP de un mismo servicio o de distintos servicios.

Los Elementos de Servicio DICOM (DIMSE, *DICOM Service Elements*) son mensajes que las AE se envían solicitando y proveyendo información. Se arman de la misma forma que las IODs, sus DEs pertenecen al grupo 0000 y se codifican usando VR. La Tabla 12 muestra los servicios DIMSE disponibles. La denominación “C” quiere decir que son para ser usados con IOD compuestos y la denominación “N” que son para IOD normalizados.

Tabla 12. Servicios DIMSE disponibles.

Nombre	Grupo	Tipo
C-ALMACENAR	DIMSE-C	operación
C-CONSEGUIR	DIMSE-C	operación
C-MOVER	DIMSE-C	operación
C-ENCONTRAR	DIMSE-C	operación
C-ECO	DIMSE-C	operación
N-REPORTE-DE-EVENTO	DIMSE-N	notificación
N-CONSEGUIR	DIMSE-N	operación
N-ASIGNAR	DIMSE-N	operación
N-ACCIÓN	DIMSE-N	operación
N-CREAR	DIMSE-N	operación
N-BORRAR	DIMSE-N	operación

El `DIMSE C-ECO` es utilizado para verificar la conectividad entre dos `AEs`. La Tabla 13 muestra el mensaje de solicitud de `C-ECO` y la Tabla 14 el mensaje de respuesta. Se puede ver que el mensaje de respuesta lleva los mismos atributos que el de solicitud más el atributo `Estado` ((0000,0900), *Status*), que lleva el valor 0000, que indica que hubo una conexión exitosa. En caso de no haber conexión no se envía mensaje de respuesta.

Tabla 13. Mensaje de solicitud de C-ECO.

Campo de Mensaje	Etiqueta	VR	VM	Descripción del Campo
Largo del Comando de Grupo	(0000,0000)	UL	1	El número par de bytes desde el final del campo de valor hasta el principio del próximo grupo.
UID Clase de SOP Afectada	(0000,0002)	UI	1	UID Clase de SOP asociada con esta operación.
Campo de Comando	(0000,0100)	US	1	Este campo distingue la operación DIMSE-C llevada por este Mensaje. A este campo se le debe asignar el valor 0030H para el Mensaje C-ECO-RQ.
ID de Mensaje	(0000,0110)	US	1	Valor especifico a la implementación que distingue este Mensaje de otros Mensajes.
Tipo de Comando de Set de Datos	(0000,0800)	US	1	Este campo indica que no hay ningún Set de Datos presente en el Mensaje y se le asignara el valor 0101H.

Tabla 14. Mensaje de respuesta de C-ECO.

Campo de Mensaje	Etiqueta	VR	VM	Descripción del Campo
Largo del Comando de Grupo	(0000,0000)	UL	1	El número par de bytes desde el final del campo de valor hasta el principio del próximo grupo.
UID Clase de SOP Afectada	(0000,0002)	UI	1	UID Clase de SOP asociada con esta operación.
Campo de Comando	(0000,0100)	US	1	Este campo distingue la operación DIMSE-C llevada por este Mensaje. A este campo se le debe asignar el valor 0030H para el Mensaje C-ECO-RQ.
ID de Mensaje al cual se le está Respondiendo	(0000,0120)	US	1	Será asignado el valor del campo ID de Mensaje (0000,0110) usado en el Mensaje C-ECO-RQ asociado.
Tipo de Comando de Set de Datos	(0000,0800)	US	1	Este campo indica que no hay ningún Set de Datos presente en el Mensaje y se le asignara el valor 0101H.
Estado	(0000,0900)	US	1	Indica el estado de la respuesta. Se le debe asignar el valor Éxito.

Cuando se desea procesar un objeto DICOM, los servicios DIMSE (comandos) se emparejan con IODs (información) para formar Pares Objeto-Servicio (SOP, *Service Object Pair*), como muestra la Figura 9. Cada SOP tiene una IOD propia, listadas en el PS3.6. En el caso de C-ECO, el atributo Tipo de Set de Datos ((0000,0800), *Data Set Type*) toma el valor de 0101, que quiere decir "vacío". El SOP Verificación (*Verification SOP*) lleva a C-ECO como componente DIMSE y tiene el componente de IOD vacío, ya que solo se busca saber si hay conexión entre AEs. Otros DIMSE llevan una IOD asociado como muestra la Figura 10. El SOP Almacenamiento (*Storage SOP*) es el encargado de enviar objetos entre AEs y se arma con C-ALMACENAR como componente DIMSE y un DS como componente IOD. El IOD ECG de 12 Derivaciones tiene un SOP Almacenamiento asociado llamado Almacenamiento de Forma de Onda de ECG de 12 Derivaciones (*12-lead ECG Waveform Storage*) cuyo UID es 1.2.840.10008.5.1.4.1.1.9.1.1.

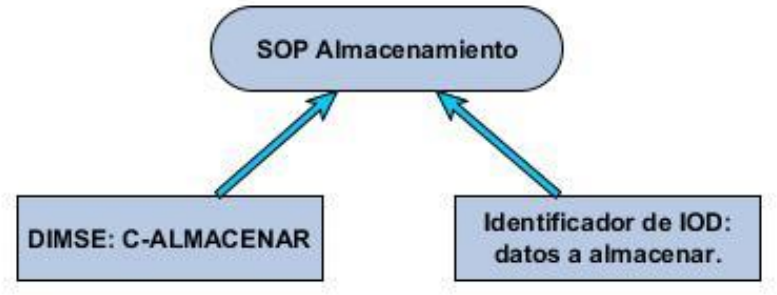


Figura 9. Los servicios DIMSE y las IOD **se juntan para formar SOP (**[6]**, Fig 7.8).**

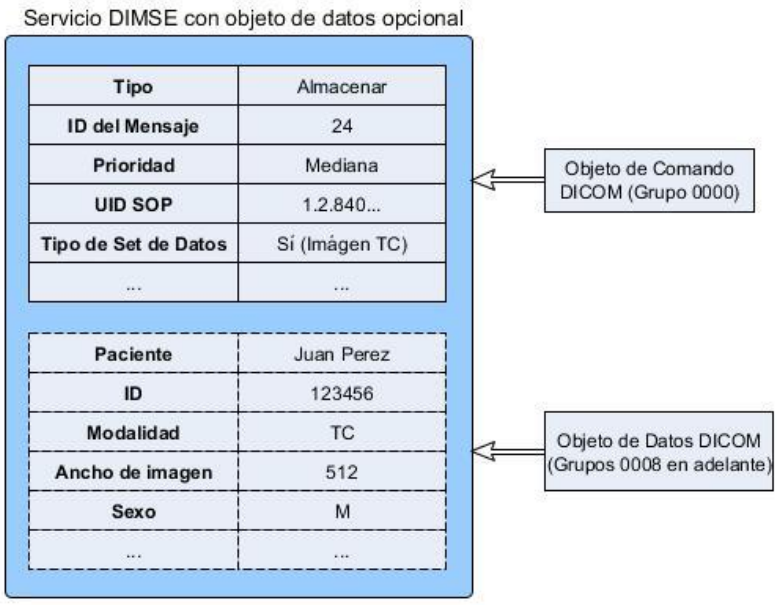


Figura 10. DIMSE con objeto asociado ([6], Fig. 7.4).

Materiales y herramientas

En este proyecto se utilizaron diversas herramientas para desarrollo y ensayos en DICOM. Muchas de estas herramientas son software de código abierto y son ampliamente usadas como bases para implementaciones privadas. Desarrollar una "librería" de DICOM propia es una tarea compleja y no se justificaba para el alcance de este trabajo. Por lo tanto, se optó por utilizar las herramientas disponibles y trabajar en base a ellas.

El lenguaje de programación usado es C++, que se eligió por ser el lenguaje en el cuales están escritas otras aplicaciones de FluxMed, facilitando una eventual integración. El entorno de desarrollo usado fue Visual Studio.

Materiales

PC: DELL Inspiron 5548

Procesador: Procesador de cuatro núcleos Intel Core i5-5200 CPU @ 2.20GHz

RAM: 8.00GB

Sistema operativo: Windows 10 Education 64 bits, version 1709

Disco rígido: SanDisk Ultra II 480GB SSD

Herramientas

Grassroots DICOM (GDCM)[3]

GDCM es una librería de software libre y multiplataforma desarrollada principalmente para leer, analizar y editar imágenes DICOM. Está publicada bajo la licencia BSD lo cual permite comercializar los desarrollos que surgen de su uso sin restricciones. La primera versión fue lanzada en el 2003 y la última versión estable (2.8.2) tiene fecha del 18 de julio 2017. Cuenta con un equipo de desarrollo y soporte activo y con un *mailing list* para consultas de usuarios y anuncios de actualizaciones. También ofrece ejemplos variados de cómo usar la librería. Está escrita en C++ y está disponible para C++, C#, Java, Python y PHP.

CMake

CMake es un software libre y multiplataforma que maneja el armado de software por medio de un método independiente de un compilador. Los proyectos de GDCM se arman con CMake, que tiene la opción de generar proyectos para Visual Studio.

[3] https://sourceforge.net/projects/gdcm/?source=navbar
https://github.com/malaterre/GDCM

Visualizadores DICOM[4]

La única manera de poder ver una imagen o forma de onda en DICOM es usando un visualizador dedicado. Hay una gran variedad de visualizadores para imágenes DICOM, tanto gratuitos como pagos. Los visualizadores para forma de onda no son tan abundantes y están limitados a visualizar archivos DICOM de ECG. Los visualizadores DICOM varían en cuán estrictos son para cumplir con el estándar. Por ejemplo, un visualizador puede tomar como archivos validos solamente a aquellos que tengan todos los atributos de tipo uno para cierta modalidad, mientras que otro visualizador puede ignorar por completo atributos de tipo uno. Para este proyecto se utilizaron tres visualizadores de ECG DICOM con el fin de corroborar los resultados.

- C# ECG Toolkit es un software para visualización de ECGs en distintos formatos con soporte para DICOM. El código fuente está disponible y es de uso libre y está publicado bajo la licencia Apache V2.0.
- ECGViewer de Pixelmed es una aplicación gratuita y escrita en Java que permite visualizar ECGs DICOM. Muestra las señales y la estructura del archivo DICOM. Está incluido dentro del Java Dicom Toolkit, que es libre y tiene el código fuente disponible. Este visualizador es el más flexible y el que más se utilizó para verificar los archivos generados.
- La empresa CharruaSoft ofrece una versión libre de su visualizador ECGViewer para ECGs DICOM.

FluxView Flux Review

El software FluxView fue utilizado para generar los archivos de texto para convertir a DICOM. El FluxReview se utilizó para contrastar visualmente los resultados de convertir los archivos originales a DICOM.

Matlab

El Image Processing Toolbox de Matlab provee algunas funciones para trabajar con imágenes DICOM. Entre ellas están *dicomread*, permite leer imágenes DICOM, *dicomwrite*, permite escribir imágenes en formato DICOM y *dicomdisp*, que muestra la estructura de un archivo DICOM de forma similar a la función de GDCM *dicomdump*.

[4] https://sourceforge.net/projects/ecgtoolkit-cs/
http://www.dclunie.com/pixelmed/software/
http://www.charruasoft.com/products/ecgv/index.php

Archivo modelo de ECG DICOM[5]

La información disponible acerca de formas de onda en DICOM fuera del estándar es escasa. Si bien es fácil conseguir ejemplos de imágenes DICOM online, es difícil encontrar ejemplos de formas de onda. Este archivo fue de gran utilidad porque permitió estudiar la estructura de una forma de onda DICOM usando las funciones *dicomdump* de GDCM y *dicomdisp* de Matlab. Esto sirvió como un gran complemento al estándar para entender cómo deben ser codificados los atributos para formas de onda. La Figura 11 muestra la visualización de este archivo con el visualizador ECGViewer de Pixelmed.

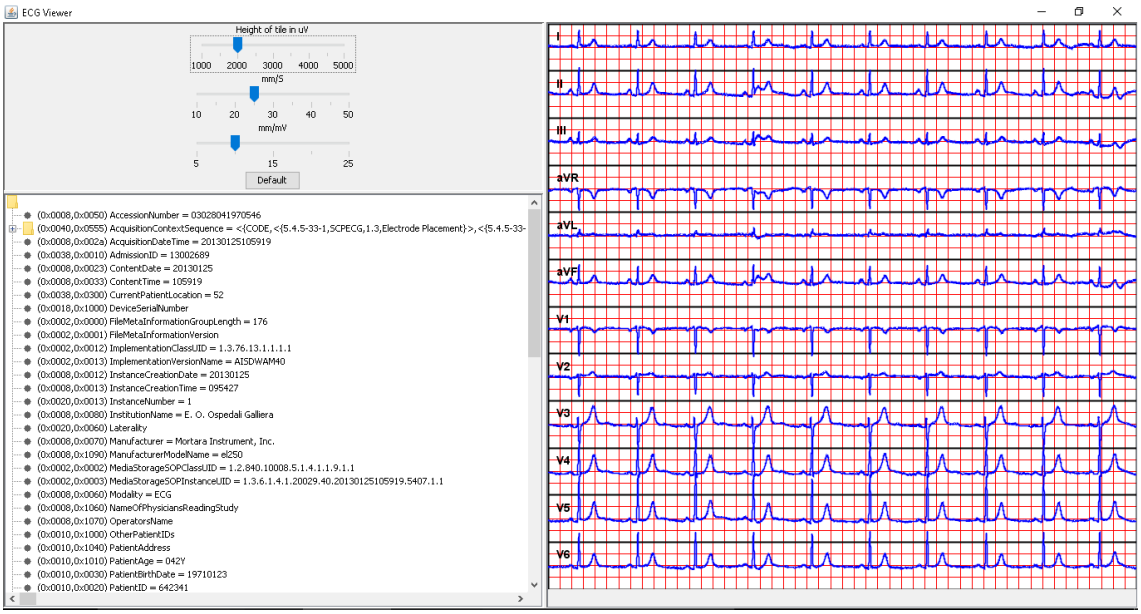


Figura 11. Visualización del archivo modelo de ECG con el ECG Viewer de Pixelmed.

Emuladores de SCU y SCP de Almacenamiento

DVTk y DICOM@OFFIS son *toolkits* para desarrollo y para validación en todas las etapas del flujo de información en DICOM. Permiten emular desde el ingreso del paciente al sistema de radiología hasta la llegada de un archivo DICOM a una estación de trabajo. Para este trabajo se utilizaron los emuladores de la clase `SOP Almacenamiento`, que sirven para enviar un archivo DICOM entre dos `AEs`. Esto se hizo con el objetivo de validar que los archivos generados por la interfaz sean transferibles en un entorno DICOM.

[5] https://github.com/marcodebe/dicomecg_convert/tree/master/sample_files

Desarrollo de interfaz DICOM

Armado de estructuras DICOM con GDCM

Archivo y Set de Datos

Para armar un archivo con GDCM se usan las clases *Writer*, *File*, *DataSet* y *FileMetaInformation*. La clase *Writer* escribe el archivo y se encarga de que cumpla con los requisitos de DICOM como, por ejemplo, que los `DEs` estén ordenados dentro del `DS` por grupo y orden creciente de atributo. La clase *DataSet* contiene al `DS` y la clase *FileMetaInformation* arma el encabezado del archivo. El Listado 1 muestra una estructura básica para armar archivos DICOM en GDCM.

```
gdcm::Writer writer;
gdcm::File &file = writer.GetFile();
gdcm::DataSet &dataset = file.GetDataSet();

//--------------------------
// Acá se arma el dataset...
//--------------------------

file.SetDataSet(dataset);

gdcm::FileMetaInformation &fmi = file.GetHeader();
fmi.SetDataSetTransferSyntax(gdcm::TransferSyntax::ExplicitVRLittleEndian;

writer.SetCheckFileMetaInformation(true);
writer.SetFile(file);

const char *outfilename = "C:\\Users\\Vladimir\\Documents\\ejemplo.dcm";

writer.SetFileName(outfilename);
writer.Write()
```

Listado 1. Inicialización de un archivo y data set en GDCM.

Elementos de Datos

Hay dos maneras de armar DEs usando GDCM: especificando la etiqueta del atributo (Listado 2) o especificándolo por su palabra clave (Listado 3). Los dos métodos se pueden usar indistintamente excepto para los atributos que tengan VRs de números en formato binario para los cuales se especifica la etiqueta correspondiente. GDCM se encarga de acomodar el largo del CV para que sea un numero par.

```
Tag tPatientName(0x10, 0x10);
gdcm::DataElement dePatientName(tPatientName);
const char PatientName[] = "Vladimir Gostuski";
dePatientName.SetByteValue(PatientName, (uint32_t)strlen(PatientName));
dataset.Insert(dePatientName);
```

Listado 2. Armado de un DE especificando la etiqueta.

```
Keywords::PatientName patientsname = { "Vladimir Gostuski" };
dataset.Insert(patientsname.GetAsDataElement());
```

Listado 3. Armado de un DE por palabra clave

UIDs

Los UIDs se generan con la "clase" *UIDGenerator*, que usa el ID raíz propio de GDCM (1.2.826.0.1.3680043.2.1143). Los UIDs ocupan un largo total de 64 bytes. Por cada conversión se genera un UID para las entidades Estudio, Serie e Instancia.

Secuencias

Para armar una secuencia en GDCM hay que:

1. Armar un DE del tipo SQ.
2. Inicializar una secuencia de ítems.
3. Armar un ítem.
4. Inicializar el DS anidado del ítem.
5. Insertar DEs en el DS anidado del ítem.
6. Insertar el ítem en la secuencia.
7. Insertar la secuencia en el DE.

El Listado 4 muestra un ejemplo de cómo armar una secuencia con otra secuencia anidada. Cada ítem también incluye un DE.

```
// Se arma el Data Element deWaveformSequence y se
// inicializa la secuencia de items WaveformSequence.
Tag tWaveformSequence(0x5400, 0x0100);
DataElement deWaveformSequence(tWaveformSequence);
deWaveformSequence.SetVR(VR::SQ);
deWaveformSequence.SetVLToUndefined();

SmartPointer<SequenceOfItems> WaveformSequence = new gdcm::SequenceOfItems();
WaveformSequence->SetLengthToUndefined();

// Se arma un item de WaveformSequence y se inicializa su data set anidado.
Item WaveformSequence_item;
WaveformSequence_item.SetVLToUndefined();
DataSet &WaveformSequence_nesteddataset = WaveformSequence_item.GetNestedDataSet();

// Se arma el data element "Sampling Frequency" y
// se agrega al data set del item de WaveformSequence.
kwd::SamplingFrequency samplingfrequency = { 256 };
WaveformSequence_nesteddataset.Insert(samplingfrequency.GetAsDataElement());

// Se arma el data element "Channel Definition Sequence" y
// se inicializa la secuencia ChannelDefinitionSequence.
DataElement deChannelDefinitionSequence(tChannelDefinitionSequence);
deChannelDefinitionSequence.SetVR(VR::SQ);
deChannelDefinitionSequence.SetVLToUndefined();

SmartPointer<SequenceOfItems> ChannelDefinitionSequence = new gdcm::SequenceOfItems();
ChannelDefinitionSequence->SetLengthToUndefined();

// Se arma un item de Channel Definition Sequence y
// se inicializa su data set anidado.
Item ChannelDefinitionSequence_item;
ChannelDefinitionSequence_item.SetVLToUndefined();
DataSet &ChannelDefinitionSequence_item_nesteddataset
= ChannelDefinitionSequence_item.GetNestedDataSet();

// Se arma el data element "Waveform Bits Stored" y
// se agrega al dataset del item ChannelDefinitionSequence.
kwd::WaveformBitsStored waveformbitsstored = { 16 };
ChannelDefinitionSequence_item_nesteddataset.Insert(waveformbitsstored.GetAsDataElement());

// Se pasa el item de Channel Definition Sequence a la secuencia ChannelDefinitionSequence.
ChannelDefinitionSequence->AddItem(ChannelDefinitionSequence_item);

// Se pasa la secuencia ChannelDefinitionSequence al data element deChannelDefinitionSequence.
deChannelDefinitionSequence.SetValue(*ChannelDefinitionSequence);

// Se pasa el data element deChannelDefinitionSequence al dataset del item de WaveformSequence.
WaveformSequence_nesteddataset.Insert(deChannelDefinitionSequence);

// Se pasa el item de WaveformSequence a la secuencia WaveformSequence.
WaveformSequence->AddItem(WaveformSequence_item);

// Se pasa la secuenica WaveformSequence al data element deWaveformSequence.
deWaveformSequence.SetValue(*WaveformSequence);
```

Listado 4. Armado de una secuencia con otra secuencia anidada.

Armado de bloques de señales

Para generar los archivos DICOM se eligió usar la IOD ECG de 12 Derivaciones, que es con el que están implementados los visualizadores. Esto permitió poder abrir el archivo y visualizar las señales graficadas durante en desarrollo de la interfaz. Todas las IODs de formas de onda son muy similares. Llegado el caso que ECG de 12 Derivaciones no sea la IOD optima, desde el punto de la programación es muy sencillo adaptar el código para que cumpla con otra IOD. La IOD ECG de 12 Derivaciones define las siguientes restricciones para los atributos pertinentes a la codificación de formas de onda:

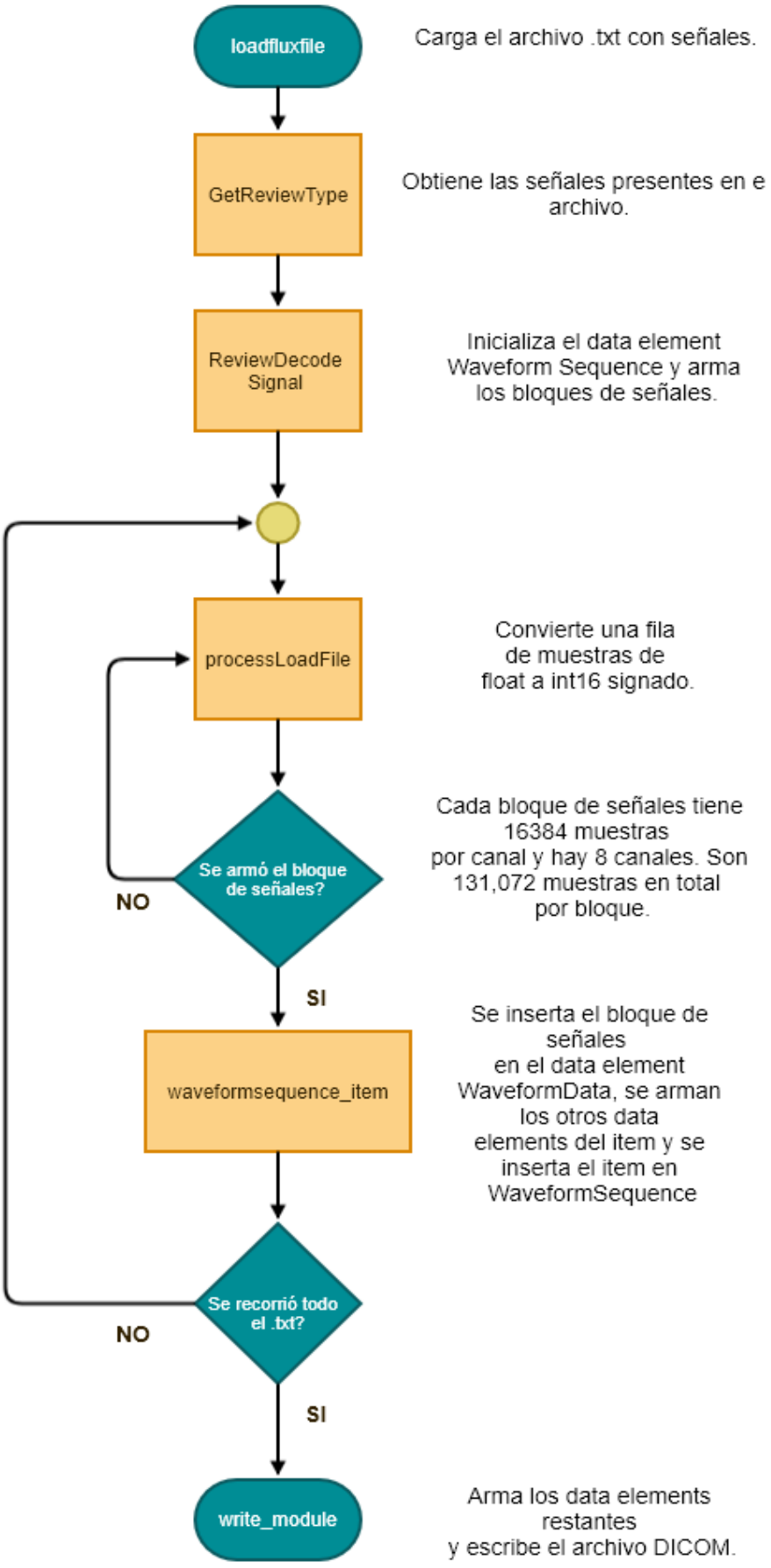
- El atributo Secuencia de Forma de Onda puede contener desde 1 hasta 5 ítems.
- El número de Canales de Forma de Onda puede ser desde 1 hasta 13 inclusive.
- El número de Numero de Muestras de Forma de Onda debe ser menor o igual a 16384.
- El valor de Frecuencia de Muestreo debe ser entre 200 y 1000 inclusive.
- El valor de Interpretación de Muestra de Formas de Onda debe ser SS.

Para armar el vector de Datos de Forma de Onda primero se busca que señales están presentes en el archivo de texto. Después se empiezan a armar los bloques de señales que irán en cada ítem de la Secuencia de Forma de Onda. Hay ocho canales por WMG y el número de muestras por canal es 16384. Por lo tanto hay 131072 (8*16384) muestras por bloque de señales. Como la frecuencia de muestreo de los monitores es de 256Hz, cada ítem contiene 64 segundos de señales.

El archivo de texto se recorre por filas, primero convirtiendo los valores de texto a double, y luego a int16 signado, el formato correspondiente para Interpretación de Muestra de Forma de Onda. Excepto la señal de volumen, todas las señales del equipo tienen un digito después del punto decimal. Una conversión directa de double a int16 haría perder el punto decimal de precisión. Para solucionar esto las señales se amplifican multiplicando por 100. El rango de int16 signado (-32768 a 32787) es suficiente para cubrir los valores que puedan tomar las señales después de ser escaladas. La señal de volumen no se amplifica ya que al convertirla no pierde precisión. Aunque se estén escalando las señales por demás, esto permite no modificar el código en el caso de que los equipos puedan adquirir señales con un digito más de resolución.

Al recorrer los archivos de texto por filas se respeta la estructura del vector para Datos de Forma de Onda, que se rellena con 131072 muestras por cada ítem de la Secuencia de Forma de Onda. Si alguna de las señales no está presente se rellena la muestra correspondiente con cero. Por lo tanto el archivo de señales originales se recorre en su totalidad, generando una única Instancia de forma de onda por archivo.

Diagrama de flujo



Resultados

Generación de archivos DICOM

Los archivos de texto generados por los equipos FluxMed se pudieron convertir a DICOM exitosamente. A continuación se muestran los resultados de generar un archivo DICOM a partir de un archivo de señales de un equipo FluxMed GrT con capnografía.

El Cuadro 2 muestra el primer segmento resultado de usar la aplicación *dicomdump* para imprimir la estructura de un archivo generado. El segmento comienza con el encabezado y luego continúa con el `DS`, empezando con los `DEs` del grupo ocho, luego los del grupo diez y veinte, hasta llegar a las secuencias `Secuencia de Contexto de Adquisición`, `Secuencia de Nombre de Concepto de Código` y `Secuencia de Concepto de Código`. El Cuadro 3 muestra el primer ítem de la `Secuencia de Forma de Onda` de forma abreviada. Al final se puede ver el atributo `Datos de Forma de Onda` que es donde se almacenan las muestras de las señales.

La Figura 12 muestra un archivo de señales FluxMed visto desde el software FluxReview y la Figura 13 la conversión a DICOM de este archivo visto desde el ECG Viewer de PixelMed. Este último visualizador solo muestra el primer ítem de la `Secuencia de Forma de Onda`, es decir 16384 muestras o 64 segundos de señales.

Los visualizadores de ECG Toolkit y Charruasoft no permitieron abrir el archivo. Haciendo un *debug* del visualizador de ECG Toolkit se encontró que éste busca valores específicos para ciertos atributos y de no cumplirse esas restricciones no permite abrir el archivo. Esto es un claro ejemplo de cómo los visualizadores en DICOM varían en cuan estrictos son para validar archivos.

```
# Dicom-File-Format
# Dicom-Meta-Information-Header
# Used TransferSyntax:
(0002,0000) UL 238                                                                  # 4,1 File Meta Information Group Length
(0002,0001) OB 00\01                                                                # 2,1 File Meta Information Version
(0002,0002) UI [1.2.840.10008.5.1.4.1.1.9.1.1]                                      # 30,1 Media Storage SOP Class UID
(0002,0003) UI [1.2.826.0.1.3680043.2.1143.5209941451377875563222547535072054774]  # 64,1 Media Storage SOP Instance UID
(0002,0010) UI [1.2.840.10008.1.2.1]                                                # 20,1 Transfer Syntax UID
(0002,0012) UI [1.2.826.0.1.3680043.2.1143.107.104.103.115.2.8.2]                   # 48,1 Implementation Class UID
(0002,0013) SH [GDCM 2.8.2]                                                         # 10,1 Implementation Version Name
(0002,0016) AE [GDCM]                                                               # 4,1 Source Application Entity Title

# Dicom-Data-Set
# Used TransferSyntax: 1.2.840.10008.1.2.1
(0008,0016) UI [1.2.840.10008.5.1.4.1.1.9.1.1]                                      # 30,1 SOP Class UID
(0008,0018) UI [1.2.826.0.1.3680043.2.1143.5209941451377875563222547535072054774]  # 64,1 SOP Instance UID
(0008,0020) DA [20180516]                                                           # 8,1 Study Date
(0008,0023) DA [20180516]                                                           # 8,1 Content Date
(0008,002a) DT [20180516004439.4]                                                   # 16,1 Acquisition Date Time
(0008,0030) TM [201805]                                                             # 6,1 Study Time
(0008,0033) TM [201805]                                                             # 6,1 Content Time
(0008,0050) SH (no value)                                                           # 0,1 Accession Number
(0008,0060) CS [ECG ]                                                               # 4,1 Modality
(0008,0070) LO [MBMed ]                                                             # 6,1 Manufacturer
(0008,0090) PN (no value)                                                           # 0,1 Referring Physician's Name
(0010,0010) PN [Vladimir Gostuski ]                                                 # 18,1 Patient's Name
(0010,0020) LO [P1]                                                                 # 2,1 Patient ID
(0010,0030) DA (no value)                                                           # 0,1 Patient's Birth Date
(0010,0040) CS [M ]                                                                 # 2,1 Patient's Sex
(0020,000d) UI [1.2.826.0.1.3680043.2.1143.76221737329428624010453262359098640 95]  # 64,1 Study Instance UID
(0020,000e) UI [1.2.826.0.1.3680043.2.1143.9203376309489349176206932933830807885]  # 64,1 Series Instance UID
(0020,0010) SH (no value)                                                           # 0,1 Study ID
(0020,0011) IS [-858993460]                                                         # 10,1 Series Number
(0020,0013) IS [1 ]                                                                 # 2,1 Instance Number
(0040,0555) SQ (Sequence with undefined length)                                     # u/l,1 Acquisition Context Sequence
  (fffe,e000) na (Item with undefined length)
    (0040,a043) SQ (Sequence with undefined length)                                 # u/l,1 Concept Name Code Sequence
      (fffe,e000) na (Item with undefined length)
        (0008,0100) SH [5.4.5-33-1-1]                                               # 12,1 Code Value
        (0008,0102) SH [SCPECG]                                                     # 6,1 Coding Scheme Designator
        (0008,0103) SH [1.3 ]                                                       # 4,1 Coding Scheme Version
        (0008,0104) LO [Electrode Placement ]                                       # 20,1 Code Meaning
      (fffe,e00d)
    (fffe,e0dd)
    (0040,a168) SQ (Sequence with undefined length)                                 # u/l,1 Concept Code Sequence
      (fffe,e000) na (Item with undefined length)
        (0008,0100) SH [5.4.5-33-1-1]                                               # 12,1 Code Value
        (0008,0102) SH [SCPECG]                                                     # 6,1 Coding Scheme Designator
        (0008,0103) SH [1.3 ]                                                       # 4,1 Coding Scheme Version
        (0008,0104) LO [Standard 12 lead positions]                                 # 26,1 Code Meaning
```

Cuadro 2. Primer segmento que incluye el encabezado DICOM y el principio del DS.

```
(5400,0100) SQ (Sequence with undefined length)                    # u/l,1 Waveform Sequence
  (fffe,e000) na (Item with undefined length)
    (003a,0004) CS [ORIGINAL]                                      # 8,1 Waveform Originality
    (003a,0005) US 8                                               # 2,1 Number of Waveform Channels
    (003a,0010) UL 16384                                           # 4,1 Number of Waveform Samples
    (003a,001a) DS [256 ]                                          # 4,1 Sampling Frequency
    (003a,0020) SH [FLUXMED ]                                      # 8,1 Multiplex Group Label
    (003a,0200) SQ (Sequence with undefined length)                # u/l,1 Channel Definition Sequence
      (fffe,e000) na (Item with undefined length)
        (003a,0208) SQ (Sequence with undefined length)            # u/l,1 Channel Source Sequence
          (fffe,e000) na (Item with undefined length)
            (0008,0100) SH [5.6.3-9-1 ]                            # 10,1 Code Value
            (0008,0102) SH [SCPECG]                                # 6,1 Coding Scheme Designator
            (0008,0103) SH [1.3 ]                                  # 4,1 Coding Scheme Version
            (0008,0104) LO [Lead I (Einthoven)]                    # 18,1 Code Meaning
          (fffe,e00d)
        (fffe,e0dd)
        (003a,0210) DS [0.1 ]                                      # 4,1 Channel Sensitivity
        (003a,0211) SQ (Sequence with undefined length)            # u/l,1 Channel Sensitivity Units Sequence
          (fffe,e000) na (Item with undefined length)
            (0008,0100) SH [uV]                                    # 2,1 Code Value
            (0008,0102) SH [UCUM]                                  # 4,1 Coding Scheme Designator
            (0008,0103) SH [1.4 ]                                  # 4,1 Coding Scheme Version
            (0008,0104) LO [microvolt ]                            # 10,1 Code Meaning
          (fffe,e00d)
        (fffe,e0dd)
        (003a,0212) DS [1 ]                                        # 2,1 Channel Sensitivity Correction Factor
        (003a,0213) DS [0 ]                                        # 2,1 Channel Baseline
        (003a,0215) DS [0 ]                                        # 2,1 Channel Sample Skew
        (003a,021a) US 16                                          # 2,1 Waveform Bits Stored
    (5400,1004) US 16                                              # 2,1 Waveform Bits Allocated
    (5400,1006) CS [SS]                                            # 2,1 Waveform Sample Interpretation
    (5400,1010) UN (OB or OW => OW)                                # 262144,1 Waveform Data
00\00\00\00\00\00\56\09\aa\f6\00\00\aa\f6\00\00\00\00\00\00\00\00\56\09\aa\f6\00\00\aa\f6\00\00\00\00\00\00\00\00\56\09\
aa\f6\00\00\aa\f6\00\00\00\00\00\00\f6\ff\4c\09\aa\f6\00\00\b4\f6\00\00        0
```

Cuadro 3. Segmento de la Secuencia de Forma de Onda con el primer ítem de esta secuencia.

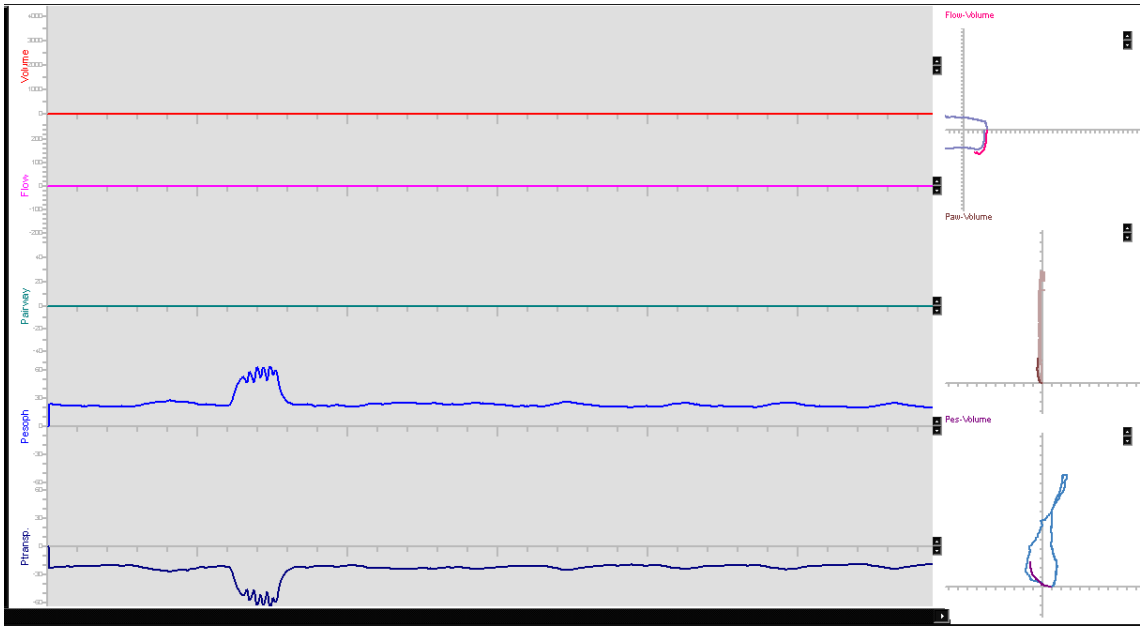


Figura 12 Archivo original visto desde FluxReview.

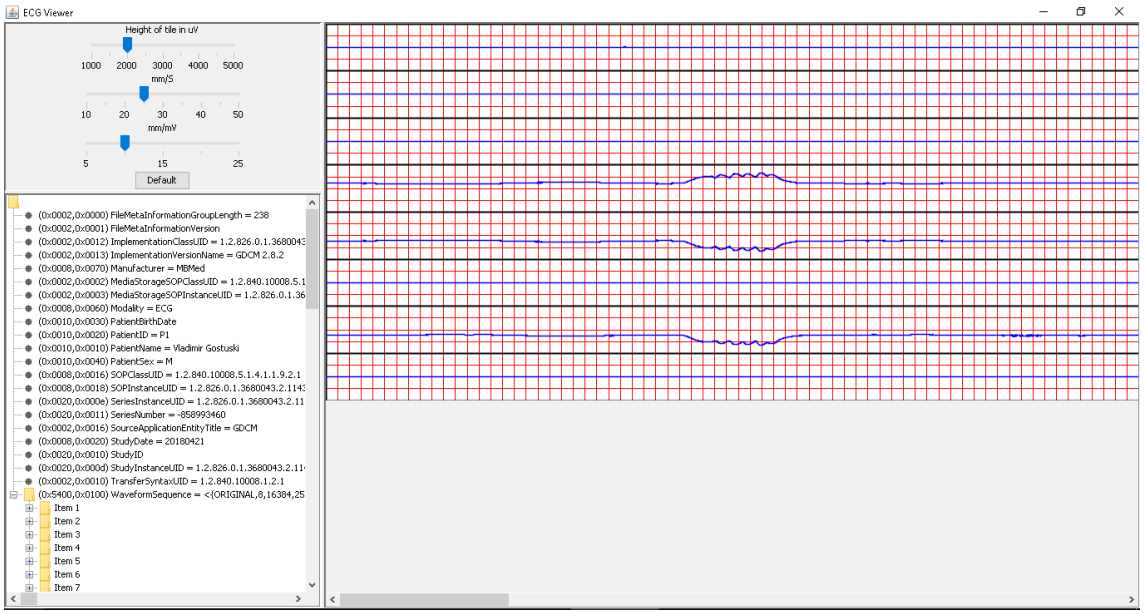


Figura 13 Archivo convertido a DICOM visto desde ECG Viewer de Pixelmed.

Ensayos con emuladores del Storage SOP

Con el archivo DICOM generado, se prosiguió a verificar si este era compatible con la clase `SOP Almacenamiento`. Para este fin se usaron los emuladores de DVTk y DICOM@OFFIS mencionados en la sección Herramientas. La Figura 14 muestra el emulador del `SOP Almacenamiento` de DVTK, con el `SCU` a la izquierda y el `SCP` a la derecha y cada uno con los valores de parámetros correspondientes.

Los parámetros para configurar en el `SCU` son `Título de AE Local`, `Título de AE Remoto`, `Dirección TCP/IP Remota` y `Numero de Puerto de AE`. Los parámetros para configurar en el `SCP` son `Título de AE Local`, `Título de AE Remoto` y `Numero de Puerto de AE`. Estos parámetros deben coincidir entre el `SCU` y el `SCP` para que se puedan encontrar e intercambiar información. La Figura 15 es el resultado de ejecutar el `SOP Almacenamiento` en DVTk, que indica que el archivo se transfirió exitosamente y sin errores o advertencias.

El Cuadro 4 muestra la ejecución desde la línea de comando del `SOP Almacenamiento` usando el `SCU` y el `SCP` de DICOM@OFFIS, que también tuvo un resultado exitoso. Por último, también se probó cruzar los `SCU` y `SCP` de cada *toolkit* para verificar que funcionen independientemente, lo que también brindó resultados exitosos.

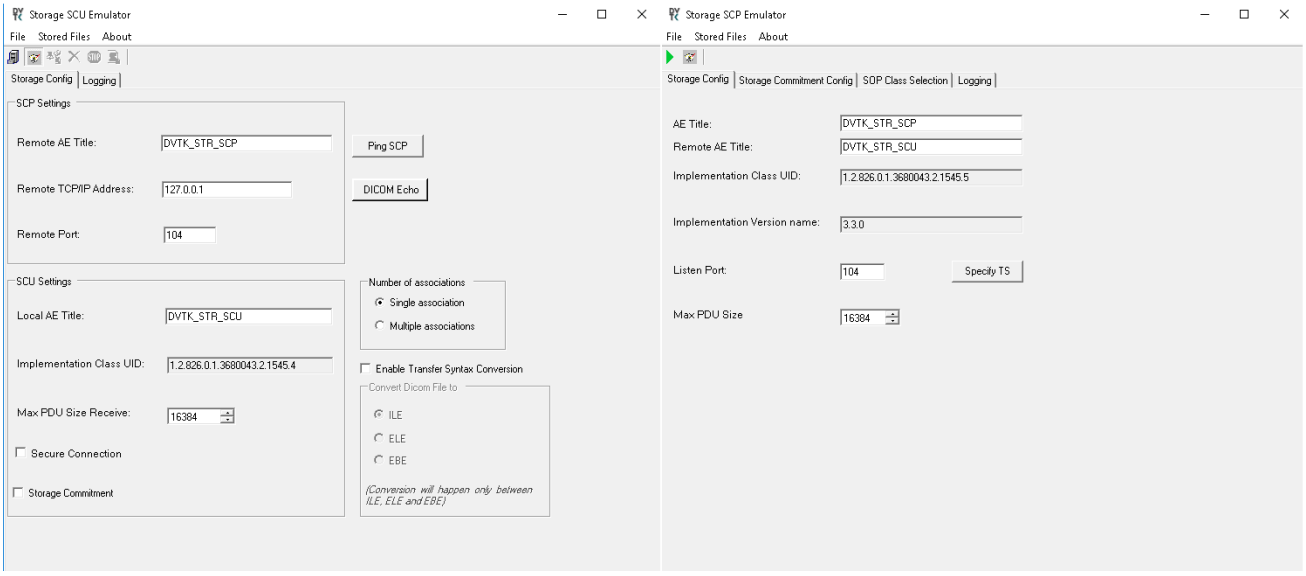


Figura 14. Emuladores de SCU y SCP del SOP Almacenamiento de DVTk.

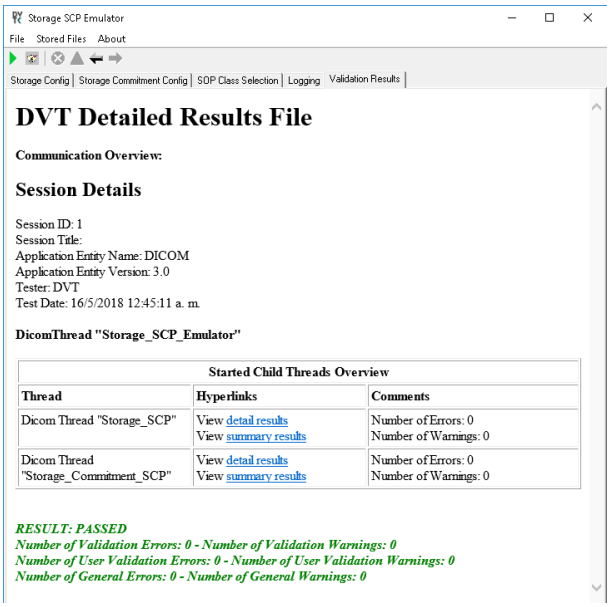


Figura 15 Resultado del Storage SOP en DVTk.

```
C:\Users\Vladimir\Documents\storescu>storescu 127.0.0.1 104 DICOM/flux.dcm

C:\Users\Vladimir\Documents\storescp>storescp -v 104
I: Association Received
I: Association Acknowledged (Max Send PDV: 16372)
I: Received Store Request (MsgID 1, TLE)
RECV:
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
I: storing DICOM file:
.\TLE.1.2.826.0.1.3680043.2.1143.5209941451377875563222547535072054774
W: DICOM file already exists, overwriting:
.\TLE.1.2.826.0.1.3680043.2.1143.5209941451377875563222547535072054774
I: Association Release
```

Cuadro 4. Ejecución del SCU y del SCP de DICOM@OFFIS desde línea de comando.

Pasos a seguir

Con este trabajo se logró generar un archivo DICOM valido y hacer un ensayo básico de su transferibilidad entre aplicaciones.

Para completar el flujo de trabajo en DICOM hace falta corroborar que los archivos generados sean almacenables en un PACS y que se puedan visualizar de forma correcta.

Para que estos sean almacenables deben ser compatibles con el PACS al cual se transfieren. Realizando una búsqueda rápida de DICOM Conformance Statements se encontró que los PACS syngo.via de Siemens, Centricity de General Electric, Intellispace de Philips, MergePACS de IBM, Impax de Agfa Healthcare, Vue de Carestream y DICOM Archive 5 de dcm4che tienen soporte para la `Clase SOP de Almacenamiento de ECG de 12 Derivaciones`. Esto da un indicio de que los archivos son almacenables, pero resta hacer pruebas en situaciones reales.

En la sección Resultados se mencionó que los archivos generados no podían ser abiertos correctamente con los visualizadores. Una solución es adaptar el software FluxReview para poder abrir archivos DICOM. Esto se podría lograr recorriendo el camino inverso a lo que hace la interfaz, es decir reconstruyendo el archivo de texto con señales a partir del archivo DICOM. Otra opción es adaptar los visualizadores de PixelMed y ECG Toolkit, para los cuales está disponible el código fuente.

Restaría también determinar si es necesario implementar una interfaz DICOM-HL7 propia para la integración con HCEs. Lo más probable es que una institución que cuente con una HCE y un PACS ya tenga estos sistemas integrados para tener disponibles estudios de diagnóstico por imágenes. Por lo tanto, hace falta estudiar mejor estas interfases para saber si la herramienta que se desarrolló puede integrarse a un flujo de trabajo ya existente sin la necesidad de una nueva interfaz. Dentro de este flujo de trabajo también se suma el ingreso de datos del paciente para ser registrados en los atributos correspondientes. Entre estos datos el de mayor importancia es el del ID del paciente, ya que un error al ingresar este dato puede causar la perdida de estudios.

Por último, para lograr interoperabilidad completa también se debe desarrollar la interoperabilidad semántica.

Conclusiones

Se pudo construir el software de interfaz de conversión de archivos generados por los monitores FluxMed a formato DICOM. Con las pruebas realizadas se comprobó que la herramienta logra los resultados esperados, con lo cual se cumplió con el objetivo de este proyecto. También se logró desarrollar una herramienta integrable a los monitores FluxMed y con un potencial valor comercial. El desarrollo de este trabajo demandó familiarizarse con DICOM, un estándar muy extenso. Si bien se trabajó con formas de onda, aprendí las bases para también trabajar con imágenes DICOM. También aprendí como es el intercambio de información entre una HCE y un centro de diagnóstico por imágenes y más específicamente como es el intercambio de información dentro de un centro de diagnóstico por imágenes. Esto llevó a entender mejor la importancia de la estandarización de la información para lograr la interoperabilidad en un contexto biomédico.

Bibliografía

[1] R. S. Dick, E. B. Steen, and D. E. Detmer, *The Computer-Based Patient Record*. 1997.

[2] F. González and D. Luna, "Manual de salud electronica," in *Publicación de las Naciones Unidas*, C. Javier and F. Andres, Eds. 2012, p. 414.

[3] A. Geraci, F. Katki, L. McMonegal, B. Meyer, and H. Porteous, *IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries*. 1991.

[4] Healthcare Information and Management Systems Society, "HIMSS dictionary of healthcare information technology terms, acronyms, and organizations." Healthcare Information and Management Systems Society, Chicago, 2013.

[5] Organización Panamericana de la Salud, *Revisión de estándares de interoperabilidad para la eSalud en Latinoamerica y el Caribe*. 2016.

[6] O. S. Pianykh, *Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide*. Springer, 2012.

[7] NEMA, *PS3.3 2017a - Information Object Definitions*. 2017.

[8] NEMA, *PS3.5 2017a - Data Structures and Encoding*. 2017.

[9] NEMA, *PS3.6 2017a - Data Dictionary*. 2017.

[10] NEMA, *PS3.10 2017a - Media Storage and File Format for Media Interchange*. 2017.