

DESCRIPCIÓN DEL ESTÁNDAR DICOM PARA UN ACCESO CONFIABLE A LA INFORMACIÓN DE LAS IMÁGENES MÉDICAS.

Description of Standard DICOM for Reliable Access To Information From Medical Images

RESUMEN

El Estándar DICOM se conforma por 18 documentos diseñados para el correcto funcionamiento e interconexión de sistemas destinados para la creación, almacenamiento, visualización, envío, recuperación, consulta, procesamiento e impresión de imágenes médicas; evitando la pérdida de información para que los usuarios accedan a ella bajo las mismas condiciones, lo cual se puede traducir en un diagnóstico acertado de una situación clínica por parte de un especialista. El objetivo de este artículo es describir los pasos a tener en cuenta para el diseño de sistemas que visualicen y procesen imágenes médicas, asegurando que los resultados obtenidos son una representación verdadera del mundo real.

PALABRAS CLAVES: Archivos DICOM, Estándar DICOM, Imágenes Médicas, Grayscale Standard Display Function, PACS.

ABSTRACT

The DICOM Standard is made up of 18 documents designed for the proper operation and interconnection of systems designed for the creation, storage, display, transmission, retrieval, consultation, processing and medical imaging, avoiding the loss of information for users to access it under the same conditions, which can be represented in an accurate diagnosis of a clinical situation by a specialist. The aim of this paper is to describe the steps to be taken into account when designing system to visualize and process medical images, ensuring that the results are a true representation of the real world.

KEYWORDS: DICOM Files, Standard DICOM, Medical Imaging Grayscale Standard Display Function, PACS.

WALTER SERNA SERNA

Ingeniero Físico,
Candidato M.Sc. Instrumentación Física
Docente Auxiliar
Universidad Tecnológica de Pereira
wserna@utp.edu.co

JUAN PABLO TRUJILLO LEMUS

Ingeniero Físico,
Candidato M.Sc. Instrumentación Física
Docente Auxiliar
Universidad Tecnológica de Pereira
jtrujillo@utp.edu.co

JORGE HERNANDO RIVERA PIEDRAHITA

Ingeniero Electrónico,
Universidad Nacional de Manizales
M.Sc. Instrumentación Física
Profesor Asociado
Universidad Tecnológica de Pereira
j.rivera@utp.edu.co

1. INTRODUCCIÓN

DICOM (Digital Imaging and Communications in Medicine) es un estándar propuesto y administrado por la National Electrical Manufacturers Association (NEMA), cuya actualización más reciente al momento de la publicación de este artículo es la del 2009 disponible en la red [1]. El propósito principal del estándar es garantizar la igualdad de condiciones desde el momento de la adquisición de un estudio imagenológico hasta el momento de ser desplegado en pantalla o impreso en papel radiográfico, después de un posible procesamiento de las imágenes.

Se debe resaltar la importancia que tienen las intensidades de grises puesto que estas deben ser las mismas sin importar el medio de visualización de la imagen, permitiendo que cualquier especialista observe lo mismo en cualquier tipo de dispositivo diagnóstico.

En la actualidad existen algunas librerías y plataformas de desarrollo de excelente calidad que insertan partes esenciales del estándar, facilitando la creación y la

implementación de algoritmos para el post-procesamiento de las imágenes médicas según la aplicación que se requiera. Esta facilidad, acompañada de la dificultad que representa el entendimiento del Estándar, obliga a la mayoría de los investigadores a confiar en los desarrolladores de dichas librerías, permaneciendo atados a las limitaciones que estas puedan presentar en cuanto a tiempo de ejecución y a la calidad de los resultados; incluyendo el hecho que son herramientas para un uso general, por lo que posiblemente no tengan fortalezas para una aplicación de un tema específico.

Regresando a las raíces, se pretende disolver los temores interpuestos por DICOM y se presenta un punto de partida para la lectura y manipulación de Imágenes médicas con base en la arquitectura de un Archivo DICOM. Así pues, inicialmente se describirá la estructura interna de dicho archivo con su respectiva interpretación que dará paso a la forma adecuada de abordar la imagen según el tamaño y el valor de los píxeles.

2. CONTENIDO

2.1 Estructura De Un Archivo DICOM

Para introducirse en el complejo entorno de la medicina, DICOM usa su propio lenguaje, basado en su modelo propio del mundo real. Se interpreta el mundo real como todos los datos físicos o descriptivos como por ejemplo el nombre del paciente, el tipo de estudio, el dispositivo médico, los parámetros de la adquisición, la imagen digital, etc. que son vistos por DICOM como elementos con sus respectivos atributos y propiedades [2]. De esta forma se establece una jerarquía entre los datos que permite realizar una clasificación según el contenido de la información por grupos, facilitando la identificación, el acceso a las variables y los parámetros de interés dentro de un mismo archivo.

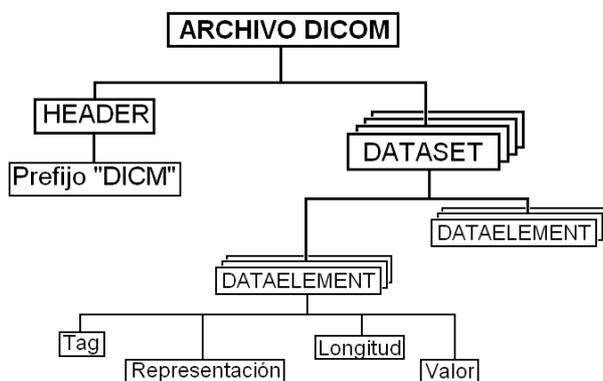


Figura 1. Estructura de un Archivo DICOM

Por lo general un archivo DICOM es reconocible por su extensión *.dcm, sin embargo esto no es una exigencia del Estándar, por lo que la forma de diferenciarlo es por medio del *HEADER* o cabecera que consta de 128 bytes de archivos de preámbulo y 4 bytes de prefijo "DICM". El preámbulo puede estar en blanco o contener información sobre la aplicación principal con la que debe ser ejecutado.

Por otra parte, el cuerpo del archivo se forma por una secuencia de *Data Sets* que representan objetos del mundo real y que a su vez están constituidos por *Data Elements* (Figura 1), que son valores codificados de los atributos del objeto, identificados y clasificados por un *TAG* o etiqueta. Cada etiqueta es un identificador único para cada *Data Element* compuesto de dos partes. El DICOM utiliza la notación siguiente para referir a una etiqueta: (gggg,eeee) el primer valor hexadecimal de 2 bytes es el número de grupo y el segundo es el número del elemento. Por ejemplo (0010,0030) corresponde a la fecha de nacimiento del paciente. Además del *TAG*, el *Data Element* está compuesto por otros tres valores: El Valor de REPRESENTACIÓN (VR) indica el tipo de dato que se tiene almacenado, la LONGITUD específica

el tamaño ocupado por el *Data Element*, y finalmente se tiene el Valor o el dato almacenado (*Value Field*).

Existe una gran variedad de *Data Elements* y no siempre estarán definidos en su totalidad dentro de un archivo, asimismo habrán *Data Elements* que no aporten información relevante para ciertas necesidades. Por esto es importante saber que como mínimo el archivo debe contener los *Data Elements* mencionados en la Tabla 1 para la adecuada lectura de la imagen:

TAG	Descripción	Tipo
(0028,0002)	Samples per Pixel	Int
(0028,0008)	Number of Frames	Int
(0028,0010)	Rows	Int
(0028,0011)	Columns	Int
(0028,0100)	Bits Allocated	Int
(0028,0101)	Bits Stored	Int
(0028,0102)	High Bit	Int
(0028,0103)	Pixel Representation	Int
(0028,1050)	Window Center	Int
(0028,1051)	Window Width	Int
(0028,1052)	Rescale Intercept	Int
(0028,1053)	Rescale Slope	Int
(7FE0,0010)	PixelData	Byte[] ó Uint16[]

Tabla 1. *Data Elements* Necesarios para la lectura adecuada de una imagen

La manera como están escritos los *Data Element* se denominan Sintaxis de Transferencia, que generalmente es igual para todos los elementos de un archivo.

La sintaxis de transferencia determina si el *Data Element* está escrito en un ordenamiento Big o Little Endian, si el valor de representación está o no incluido en el elemento (VR explícito o Implícito) y el tipo de compresión de la imagen (Mapa de Bits, JPEG o diferentes tipos de compresión).

Los datos del píxel se pueden enviar en un formato nativo o en un formato encapsulado definido fuera del estándar DICOM. Si están enviados en un formato nativo, las muestras del píxel se codifican como el encadenamiento directo de los bits de cada muestra del píxel. Si los bits están almacenados en un formato encapsulado, las muestras del píxel se codifican según el proceso de codificación definido por uno de las sintaxis de transferencia. Como el objetivo del artículo es brindar una introducción al Estándar DICOM, se asumirá que las imágenes manipuladas vienen en un formato nativo, es decir sin encapsular.

2.2 Lectura Adecuada del PixelData

Según la Tabla 1, los valores de los píxeles de la imagen se almacenan en una cadena de Bytes o de Uint16 que corresponde al *Data Element* (7FE0,0010) llamado *PixelData*. Ahora, con la información de los otros *Data Elements* esta cadena se irá transformando a través de varios pasos en una matriz que representará a la imagen con los valores finales de los píxeles, según las necesidades del usuario y sin pérdida de información. El siguiente procedimiento es una propuesta de los autores y se aclara que puede existir un orden diferente de los factores que permita llegar a un mismo resultado.

Inicialmente es necesario conocer cuántos elementos (muestras) consecutivos de la cadena forman el valor de un píxel. Esto es, si la imagen está en escala de grises solo se requiere un valor que represente la intensidad del píxel, o si la imagen está a color (RGB) entonces se requieren tres valores, uno para el canal Rojo, otro para el Verde y otro para el Azul. Dentro del *PixelData* es imposible determinar el conjunto de valores consecutivos que conforman un píxel, por esto se utiliza el *Data Element* (0028,0002) *Samples per Pixel*, que indica el número de canales de la imagen o el número de muestras por píxel, y según lo mencionado puede tomar dos valores: Será igual a 1 si es una imagen en escala de grises o será igual a 3 si es una imagen en RGB.

El siguiente paso permitirá definir si la cadena debe ser de tipo Byte o Uint16, gracias al *Data Element* (0028,0100) *Bits Allocated* (*BA*), que implícitamente define el rango de valores que puede tomar cada elemento de la cadena. Por ejemplo si el número de *BA* es *N* el rango será $0:(2^N - 1)$. Asimismo, según el estándar los bits permitidos podrán ser igual a 8, 10, 12 ó 16. Si se permiten 8 bits por elemento, entonces la cadena podrá ser de tipo Byte, por otra parte si se permiten 10,12 ó 16 bits por elemento la cadena deberá ser tipo Uint16. Este paso es importante pues si se almacenan valores de 16 bits en una variable tipo Byte se perderá información, y si se almacenan valores de 8 bits en una variable tipo Uint16 se desperdiciará espacio en memoria.

Por otra parte, no siempre se utilizan la totalidad de los Bits permitidos por elemento, una de las razones puede ser que aún no existan equipos diagnósticos que capturen la imagen con tal detalle que diferencie entre 2^{16} (65536) niveles de grises, ni pantallas capaces de visualizarlos (en la actualidad existen pantallas con una profundidad de 2^{10} niveles de intensidad), asimismo el ojo humano solo nota cambios definidos en un rango de 0-100 niveles de grises. Sin embargo, estos amplios rangos que no son diferenciables para el ojo pero que indudablemente entregan un mayor detalle de la escena, pueden ser muy útiles en la aplicación de algoritmos matemáticos.

Volviendo al tema, El *Data Element* (0028,0101) *Bits Stored* (*BS*), es el que determina el número de bits usados

dentro de los bits permitidos, por lo que su valor siempre será menor al de *BA*, y será el mismo para todos los elementos de la cadena al igual que los siguientes *Data Elements*. No necesariamente los bits utilizados se ubican en las posiciones iniciales dentro de los permitidos, para poder localizarlos se usa el *Data Element* (0028,0102) *High Bit* (*BH*), que contiene la posición del último bit usado contando desde cero. Así pues, se da lugar a un offset que no será lineal en el valor final del píxel pues se presenta en la posición dentro de la cadena (Figura 2).

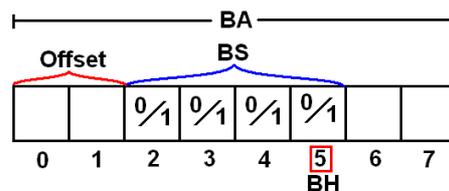


Figura 2. Representación de la posición de *Bits Stored* dentro de los *Bits Allocated*

La ecuación 1 es una forma de determinar el offset a partir de *Bits Stored* y de *High Bit*:

$$Offset = (BH + 1) - BS \quad (1)$$

Se suma la unidad a *BH* debido a que las posiciones se cuentan desde cero. Una vez se conozca el offset se procede a la primera modificación de los elementos de la cadena, que como se mencionó no es una relación lineal:

$$valor = valor \times 2^{-Offset} \quad (2)$$

Donde *valor* representa cada uno de los elementos de la cadena. Por lo general el offset siempre es igual a cero, por lo que no habría razón para aplicar la ecuación 2 a la cadena, sin embargo se aconseja la implementación de este paso para garantizar la robustez del sistema.

Debido a que una imagen médica puede representar magnitudes físicas, sus píxeles pueden tomar valores positivos o negativos; por ejemplo un estudio de Resonancia Nuclear Magnética en secuencia de Contraste de Fase, donde la intensidad del píxel es directamente proporcional a la velocidad de un fluido bidireccional y perpendicular al corte anatómico [4].

El *Data Element* (0028,0103) *Pixel Representation*, puede tomar dos valores: 0 que indica que todos los elementos son positivos, ó 1 que indica que pueden haber tanto elementos positivos como negativos. Sin embargo, recordando que la cadena es de tipo Byte o Uint16 se puede ver que no es posible almacenar valores negativos, por lo que en caso de que *Pixel Representation* sea igual a 1, el Estándar recurre al *Complemento a 2*, que codifica los elementos con base en la ecuación 3:

$$valor = \begin{cases} valor & ,si\ valor \leq \frac{2^{BA}}{2} \\ valor - 2^{BA} & ,si\ valor > \frac{2^{BA}}{2} \end{cases} \quad (3)$$

Donde *BA* corresponde al número de *Bits Allocated*. De esta forma se genera una nueva cadena que contiene cantidades con signo. Se aclara que esta cadena sirve para definir magnitudes físicas pero se debe tener en cuenta que los valores de intensidad de los píxeles no pueden ser negativos.

Para culminar con la conversión de cadena a matriz y de esta forma conocer la ubicación final de los píxeles dentro de la imagen se usan los *Data Elements* (0028,0010) *Rows* y (0028,0010) *Columns*, que almacenan las cantidades de filas y de columnas respectivamente de la matriz en píxeles. Lo que se debe hacer es dividir la cadena en un número igual a la cantidad de columnas, teniendo en cuenta que cada una de estas subcadenas contendrá un total de elementos igual al número de filas. Así pues, si se desea ubicar la posición de un píxel en términos de coordenadas horizontales y verticales, se puede hacer ubicando la subcadena que corresponda a la coordenada vertical, para posteriormente ubicar la posición que corresponda a la coordenada horizontal dentro de la subcadena. Es importante tener en cuenta el número de elementos por píxel.

Hasta el momento se ha extraído toda la información espacial de la imagen. Sin embargo para poder desplegarla en pantalla también se deben modificar los valores de intensidad del píxel que han sido codificados según las especificaciones de los fabricantes de los equipos diagnósticos, para garantizar que en todos los dispositivos se pueda ver una misma información.

Grayscale Standard Display Function [6] está basada en la sensibilidad de contraste del ojo humano, el cual es más sensible en las áreas brillantes de una imagen que en las áreas oscuras. Esta variación hace más fácil distinguir los cambios de intensidad en zonas brillantes que en las zonas oscuras [2, 6]. Dicho esto se puede intuir que el modelo no es lineal, por lo que DICOM ha definido tablas basadas en el modelo del sistema de visión humano desarrollado por Barten [7, 8] que permite comparar los valores almacenados en el archivo para obtener unos nuevos valores según el rango de luminancia de la pantalla. Sin embargo, aunque el modelo no es lineal, en muchos casos se aproxima a una línea recta, facilitando el algoritmo ya que no se necesita una tabla, sino la pendiente y la intercepción con un eje para generar una ecuación lineal. Esta primera transformación sobre los valores de los píxeles se conoce dentro del Estándar DICOM como *Modality LUT IE*.

Para poder hacer un ajuste en la escala de grises, se plantea un sistema cartesiano donde el eje X representará

los píxeles almacenados en el archivo DICOM codificados según el Modelo de Barten, y el eje Y representará los píxeles que serán desplegados en pantalla.

Inicialmente se construirá la ecuación de la recta que permite encontrar un píxel de salida *y* a partir de un píxel almacenado *x*, como se muestra en la ecuación 4 y en la Figura 3:

$$y = mx + b \quad (4)$$

Donde *b* será la intercepción de la recta con el eje X y *m* será la pendiente de la recta. Estos dos valores están almacenados en los *Data Elements* (0028,1052) *Rescale Intercept* y (0028,1053) *Rescale Slope* respectivamente. Si por algún motivo no se encuentran estos *Data Elements* dentro del archivo, se debe asumir que la intercepción de la recta se efectúa en el origen, y que la pendiente de esta es igual a 1. Esto quiere decir que los valores de entrada serán igual a los de salida, por lo que la ecuación no modificará los píxeles almacenados.

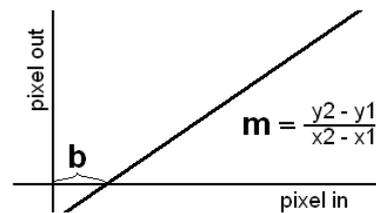


Figura 3. Ajuste de los valores de los píxeles por medio de los *Data Elements Rescale Intercept* y *Rescale Slope*

Una vez culminada la graduación de la pendiente y del intercepto, se aplica la última transformación sobre los píxeles entregados por el *Modality LUT*, que tendrá como resultado los píxeles válidos para ser impresos o desplegados en pantalla. Esta transformación se conoce como *VOI LUT IE*, y tiene como objetivo establecer la profundidad o el rango final de la intensidad de grises. Lo que se pretende es limitar la línea recta generada anteriormente, pues los valores de los píxeles deben tener un máximo y un mínimo. Adicionalmente se podrá modificar el contraste y el brillo de la imagen según las necesidades del usuario final.

Igual que en la transformación anterior, existe un modelo no lineal y otro lineal. Siempre que los *Data Elements* (0028,1050) *Window Center* (0028,1051) *Window Width* estén definidos, se trabaja con el modelo lineal que obedece a la ecuación 5:

$$y = \begin{cases} y_{min} & , si\ x \leq c - 0.5 - \frac{w-1}{2} \\ y_{max} & , si\ x > c - 0.5 + \frac{w-1}{2} \\ \left(\frac{x - (c - 0.5)}{w - 1} + 0.5 \right) (y_{max} - y_{min}) + y_{min} & , otro\ caso \end{cases} \quad (5)$$

Donde x es el pixel a modificar, y el pixel a mostrar en pantalla, c es el *Window Center*, w es el *Window Width* y $y_{max} - y_{min}$ es el nuevo rango en el que se calculará la escala de grises. En la Figura 4 se puede ver la línea recta transformada y con los respectivos límites. A todos los valores fuera del rango se les asigna el valor del límite superior o el inferior según sea el caso.

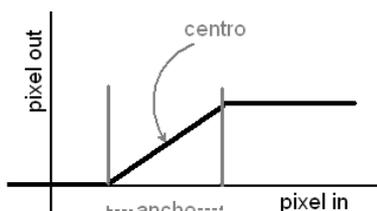


Figura 4. Ajuste final de los valores de los píxeles para su representación en pantalla.

Finalmente se presenta el diagrama de bloques para desplegar en pantalla una imagen médica de un archivo DICOM

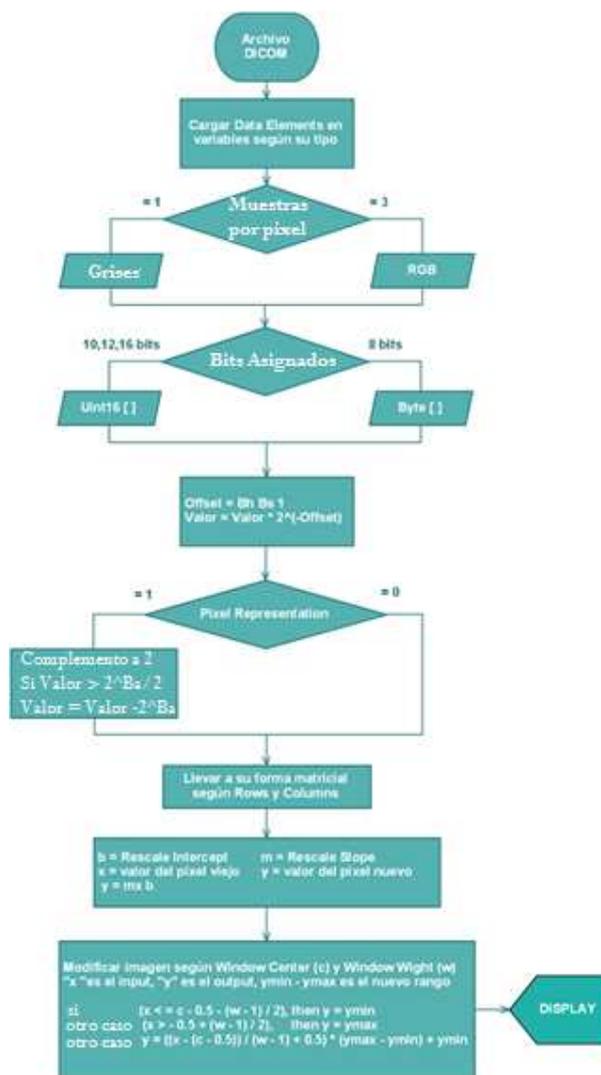


Figura 5. Diagrama de Bloques para abrir un Archivo DICOM

3. CONCLUSIONES Y RECOMENDACIONES

A pesar de que existen muchas herramientas que facilitan la manipulación de archivos DICOM, es primordial como desarrollador conocer su estructura para analizar la eficiencia de los algoritmos implementados y no depender indirectamente de otro tipo de sistemas.

Se recomienda tener siempre presente la procedencia de los estudios de imágenes médicas analizadas, pues los *Data Elements* pueden variar de un equipo a otro. Esto se encuentra en las especificaciones del fabricante.

4. BIBLIOGRAFÍA

[1] Página oficial de DICOM, <http://medical.nema.org/>

- [2] OLEG S. PIANYKH, Digital Imaging and Communications in Medicine (DICOM) A Practical Introduction and survival Guide.
- [3] NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION, Digital Imaging and Communications in Medicine (DICOM) Part 3: Information Object Definitions.
- [4] KORNIENKO, V. PRONIN, I, Diagnostic Neuroradiology, Burdenko Neurosurgical Institute Dept. of Neuroradiology, 2009.
- [5] NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION, Digital Imaging and Communications in Medicine (DICOM) Part 6: Data Dictionary
- [6] NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION, Digital Imaging and Communications in Medicine (DICOM) Part 14: Grayscale Standard Display Function
- [7] BARTEN, P.G.J., Physical model for the Contrast Sensitivity of the human eye. Proc. SPIE 1666, 57-72 (1992)
- [8] BARTEN, P.G.J., Spatio-temporal model for the Contrast Sensitivity of the human eye and its temporal aspects. Proc. SPIE 1913-01 (1993)