

## Capítulo 5

---

# El Estándar DICOM

[10] [12] [13] [14] [15] [19]

### 5.1. Introducción

Antes de continuar con el desarrollo del proyecto, es bueno que dediquemos un capítulo a realizar un estudio en profundidad del estándar de *DICOM*<sup>1</sup>, especialmente de los tipos de servicio de almacenamiento (*C-STORE*) y verificación (*C-ECHO*) que son básicamente los únicos que necesitan utilizar las aplicaciones cliente que se han desarrollado. El servidor será quien suministre dichos tipos de servicio a los clientes.

Con la introducción de distintas técnicas de diagnóstico mediante imágenes digitales en los años 70, y el creciente uso de los ordenadores en aplicaciones clínicas, el Colegio Americano de Radiología (ACR, *American College of Radiology*), y la Asociación Nacional de Fabricantes Eléctricos (NEMA, *National Electrical Manufacturers Association*) vieron la necesidad de estandarizar un método para la transferencia de imágenes y de la información asociada a ellas entre dispositivos de

---

<sup>1</sup> Digital Imaging and Communications in Medicine

distintos fabricantes, los cuales pueden producir imágenes en distintos formatos digitales.

Con tal fin, en 1983 se creó un comité, formado por la ACR y la NEMA, con el objetivo de desarrollar un estándar que:

- ♣ Promoviese la comunicación de imágenes digitales, independientemente del fabricante del equipo.
- ♣ Facilitase el desarrollo y expansión de Sistemas de Almacenamiento y Comunicación de Imágenes (PACS, *Picture Archiving and Communications Systems*) de manera que puedan también interactuar con otros sistemas presentes en el hospital.
- ♣ Permitir la creación de bases de datos de diagnósticos que puedan ser consultadas por una gran variedad de dispositivos distribuidos Geográficamente.

En 1985 se publica la primera versión, la 1.0, a la que siguieron dos revisiones, la primera en Octubre de 1986, y la segunda en Enero de 1988. Ese mismo año se publica la segunda versión, la 2.0. Dicha versión incluye la 1.0, las revisiones mencionadas además de otras revisiones adicionales que proporcionaban soporte para dispositivos de visualización, introducía un nuevo esquema jerárquico para la identificación de imágenes y añadía nuevos elementos de información para una mejor descripción de los distintos tipos de imágenes.

En la actualidad, el Estándar DICOM va por la versión 3.0, e incluye numerosas mejoras sobre las versiones anteriores:

- ♣ *Es aplicable a entornos en red.* Las versiones anteriores estaban orientadas sobre todo a conexiones punto a punto, y era necesaria una interfaz adicional de red (NUI, *Network Unit Interface*) para conectar en red los equipos. En la nueva versión, DICOM soporta operaciones en red utilizando estándares de la industria, como OSI y TCP/IP.
- ♣ *Especifica Clases de Servicio.* En versiones anteriores se hacía referencia únicamente a la transferencia de datos. Con la nueva versión se definen las

*Clases de Servicio*, que permiten definir la semántica de los distintos comandos y de la información asociada.

- ♣ *Especifica Niveles de Conformidad*. Se define la forma en que cada fabricante debe proporcionar una *Declaración de Conformidad*, donde se diga qué partes del Estándar se utilizan en un determinado equipo o implementación.
- ♣ *Está estructurado en partes*. Esto permite una rápida evolución del Estándar en un entorno donde se produce una evolución continua. Para la estructuración en partes se han seguido las directivas de la ISO pertinentes.
- ♣ *Introduce nuevos Objetos de Información*. No sólo para las imágenes y gráficas, sino también para estudios, informes, etc.
- ♣ *Identifica de forma única los Objetos de Información*. Utilizando técnicas de identificación única, se facilita la definición biunívoca de relaciones entre *Objetos de Información* a medida que se propagan por la red.

## 5.2. Estructura del Estándar DICOM

El Estándar DICOM se ha estructurado en varias partes, siguiendo las directivas de la ISO al respecto. Las partes más importantes de que se compone son las siguientes:

- ♣ PS 3.1: Introducción
- ♣ PS 3.2: Conformidad
- ♣ PS 3.3: Definiciones de Objetos de Información
- ♣ PS 3.4: Especificaciones de las Clases de Servicio
- ♣ PS 3.5: Estructuras de Datos y Codificación
- ♣ PS 3.6: Diccionario de Datos
- ♣ PS 3.7: Intercambio de Mensajes
- ♣ PS 3.8: Soporte de Red para el Intercambio de Mensajes
- ♣ PS 3.9: Soporte de Comunicaciones Punto a Punto para el Intercambio de Mensajes

Todas las partes del Estándar están relacionadas pero son independientes entre sí. Podemos ver las relaciones entre ellas en la figura 5.1.

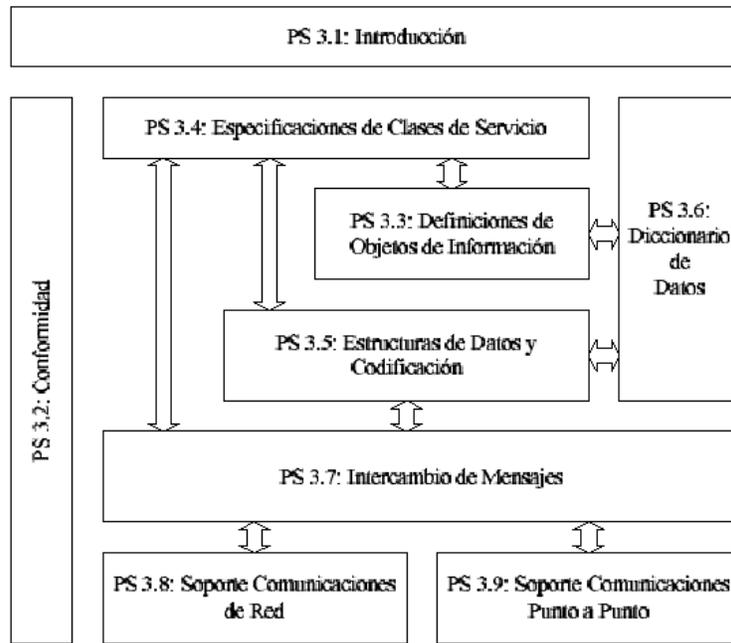


Figura 5.1. Relaciones entre las distintas partes del estándar DICOM

### 5.3. Modelo de información DICOM

El Modelo de Información DICOM define la estructura y organización de la información relacionada con la comunicación de imágenes médicas. En la figura 5.2 podemos ver las relaciones entre las principales estructuras del modelo.

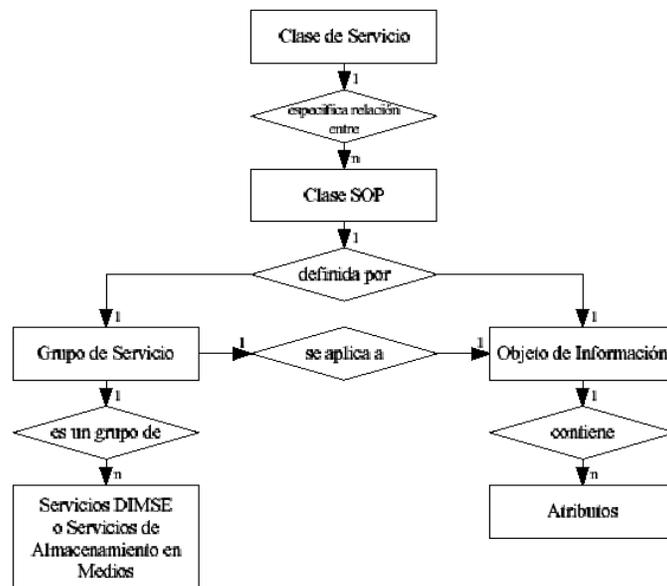


Figura 5.2. Modelo de información DICOM

### 5.3.1. Objetos de Información

Un *Objeto de Información* (IOD, *Information Object Definition*) es un modelo abstracto de datos utilizado para especificar información acerca de objetos del mundo real. Con los IODs, las *Entidades de Aplicación* van a tener un punto de vista común acerca de la información que están intercambiando.

Un Objeto de Información no representa una determinada instancia de un objeto del mundo real, sino más bien una clase de objetos del mundo real que comparten las mismas propiedades. Los Objetos de Información que se utilizan para representar una clase simple de objetos del mundo real se denominan *Objetos de Información Normalizados* (*Normalized IOD*). Los Objetos de Información que incluyen información de varios objetos relacionados entre sí se denominan *Objetos de Información Compuestos* (*Composite IOD*).

### 5.3.2. Atributos

Los *Atributos* de un Objeto de Información describen las propiedades de una instancia de un objeto del mundo real. Los Atributos que están relacionados entre sí se agrupan en *Módulos*.

Los Atributos se codifican en los llamados *Elementos de Datos* utilizando las reglas y los conceptos de Representación y Multiplicidad de Valor recogidos en el Estándar DICOM, y que veremos un poco más adelante (ver sección 5.7).

### 5.3.3. Servicios DIMSE y Servicios de Almacenamiento en Medios

Los *Servicios DIMSE* permiten a las Entidades de Aplicación DICOM la comunicación en línea, mediante la invocación de operaciones o notificaciones a través de una red o de una interfaz punto a punto. Hay dos tipos de Servicios DIMSE:

- ♣ *DIMSE-C*: aplicables sólo a Objetos de Información Compuestos. Proporcionan únicamente servicios de operación.

- ♣ *DIMSE-N*: aplicables sólo a Objetos de Información Normalizados. Proporcionan servicios tanto de operación como de notificación.

Los Servicios de Almacenamiento en Medios permiten a las Entidades de Aplicación DICOM el intercambio de información utilizando operaciones relacionadas con medios de almacenamiento.

#### **5.3.4. Grupo de Servicios DIMSE**

Un *Grupo de Servicio DIMSE* especifica una o más operaciones / notificaciones que son aplicables a un determinado Objeto de Información. Estos Grupos de Servicios se encuentran definidos en las correspondientes especificaciones de las Clases SOP.

#### **5.3.5. Clases SOP**

Una *Clase Par Objeto-Servicio (Clase SOP)* se define mediante la unión de un Objeto de Información y de un Grupo de Servicio DIMSE. La definición de la clase contendrá reglas e información semántica que restringirá el uso de los servicios del Grupo de Servicio DIMSE y/o los Atributos del Objeto de Información.

DICOM define dos tipos de Clases SOP, *Normalizadas* o *Compuestas*. Las Clases SOP Normalizadas se definen como la unión de un Objeto de Información Normalizado y un conjunto de Servicios DIMSE-N. Las Clases SOP Compuestas se definirán entonces como la unión de un Objeto de Información Compuesto y un conjunto de Servicios DIMSE-C.

#### **5.3.6. Clases de Servicio**

Una *Clase de Servicio* define un grupo de una o más Clases SOP relacionadas con una determinada función. También define las normas que deben seguir las implementaciones para mantener un cierto nivel de conformidad con respecto a una o más Clases SOP, ya sea como *Usuarías de la Clase de Servicio (SCU, Service Class User)* o como *Proveedoras de la Clase de Servicio (SCP, Service Class Provider)*.

## 5.4. Modelo DICOM del mundo real

En la figura 5.3 podemos ver una parte del *Modelo DICOM del Mundo Real*, el cual identifica los objetos más relevantes del mundo real y las relaciones entre ellos, siempre dentro del ámbito de DICOM. El objetivo es proporcionar un marco de trabajo común para asegurar la consistencia entre los distintos Objetos de Información definidos en el Estándar.

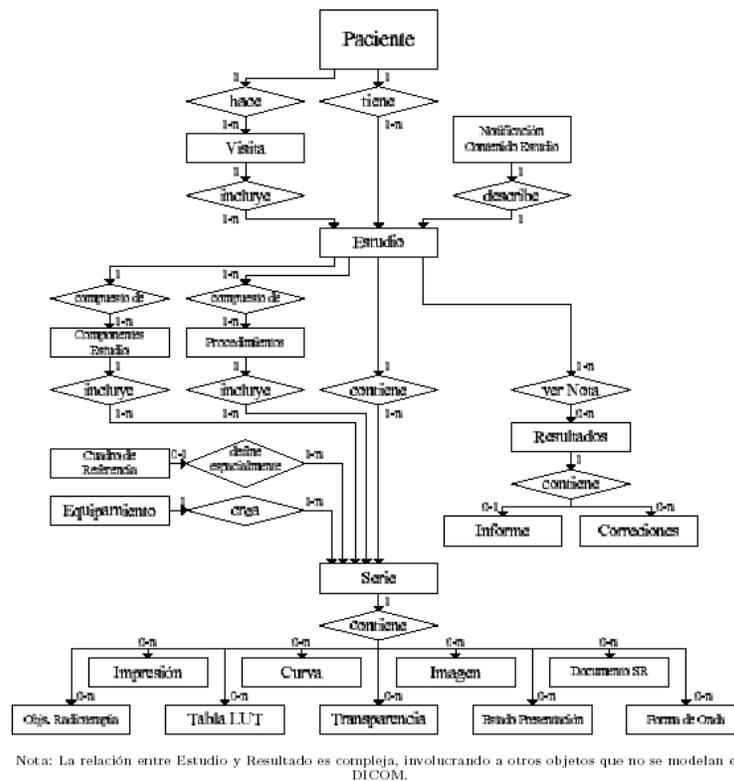


Figura 5.3. Modelo DICOM del mundo real

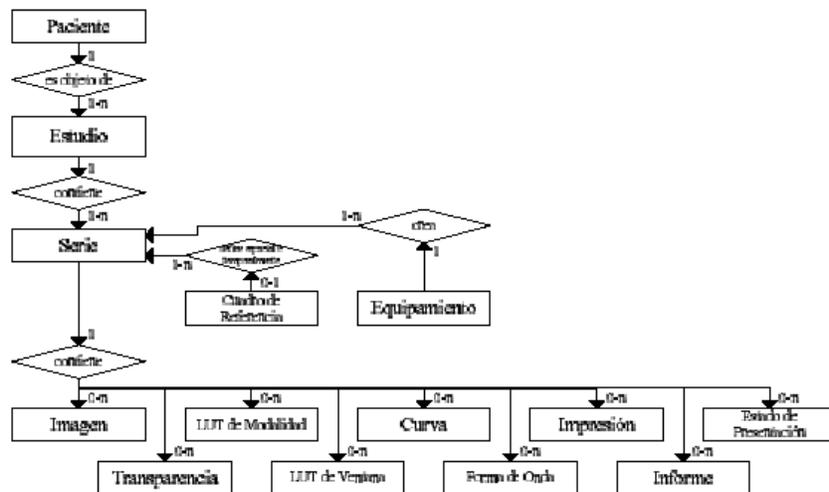
## 5.5. Objetos de Información

### 5.5.1. Objetos de Información Compuestos

Para este tipo de objetos DICOM proporciona el Modelo Entidad - Relación que describe las relaciones de los componentes o *Entidades de Información* (IE, *Information Entities*) del Objeto de Información especificado. Este modelo proporciona un contexto

completo de cómo ha de interpretarse la información cuando ésta se intercambia entre distintas Entidades de Aplicación DICOM.

El Modelo Entidad-Relación que podemos ver en la figura 5.4 es común para todos los Objetos de Información Compuestos, si bien en general cada uno utilizará un subconjunto de éste, de forma que quede definido con mayor precisión el contexto. Las Entidades de Información que integran el Modelo Entidad-Relación son las siguientes:



**Figura 5.4.** Modelo Entidad-Relación de los Objetos de Información Compuestos

- ♣ *Paciente*: define las características del paciente que está siendo objeto de un estudio.
- ♣ *Estudio*: define las características del estudio realizado a un paciente. Es una colección de una o más series de Imágenes, Estados de Presentación, Documentos, Transparencias, y/o Curvas lógicamente relacionadas cuyo objetivo es el diagnóstico del paciente.
- ♣ *Serie*: define los Atributos utilizados para agrupar instancias relacionadas entre sí.
- ♣ *Equipamiento*: describe al dispositivo particular que ha producido una serie de instancias, pero sin entrar en mucho detalle acerca de los parámetros utilizados para generar las imágenes.
- ♣ *Cuadro de Referencia*: describe el sistema de coordenadas que contiene la información espacial y temporal de una serie de instancias. Cuando está

presente, permite relacionar espacial o temporalmente distintas series de instancias entre sí.

- ♣ *Imagen*: define los atributos que contienen la información gráfica (los pixels) de la imagen.
- ♣ *Transparencia*: define los atributos que describen un conjunto independiente de Planos de Transparencia. Estos planos pueden contener información en forma de mapa de bits, gráficos o texto y se utilizan para indicar en la imagen zonas de interés, puntos de referencia o insertar anotaciones.
- ♣ *Curva*: representa curvas mediante una sucesión de puntos unidos entre sí. Se pueden utilizar como una imagen en sí, sin necesidad de información gráfica adicional, aunque lo más habitual es que se utilicen para especificar gráficos multidimensionales, regiones de interés y anotaciones.
- ♣ *LUT de Modalidad*: describe las transformaciones necesarias para convertir la información gráfica generada por un dispositivo a información gráfica independiente del equipo.
- ♣ *LUT de Ventana*: describe las transformaciones necesarias que se deben aplicar a la información gráfica de una imagen para obtener otra con algún interés especial para visualizarla, imprimirla, etc.
- ♣ *Estado de Presentación*: define como debe representarse una imagen de referencia en un dispositivo independiente en escala de grises, y qué anotaciones gráficas y transformaciones espaciales y de contraste han de aplicarse a la imagen de referencia.
- ♣ *Forma de Onda*: representa una forma de onda multicanal muestreada. La forma de onda consiste en la medida de alguna cantidad física (voltaje eléctrico, presión, concentración de gas, sonido, etc.), muestreada a intervalos de tiempo regulares.
- ♣ *Informe*: describe el contenido de un documento de Informe Estructurado.

### 5.5.2. Objetos de Información Normalizados

Los Objetos de Información Normalizados constan de una única Entidad de Información, que puede estar compuesta a su vez por varios Módulos. A diferencia de los Objetos de Información Compuestos, no es necesario un Modelo Entidad-Relación,

ya que los Objetos de Información Normalizados representan una clase simple de objetos del Mundo Real, por lo que no hay relaciones que modelar.

## **5.6. Clases de Servicio**

En esta sección vamos a ver las especificaciones de algunas de las Clases de Servicio definidas en el Estándar DICOM que se han utilizado para la realización de este Proyecto.

### **5.6.1. Clase de Servicio de Verificación**

La *Clase de Servicio de Verificación* define un servicio que verifica la comunicación entre Entidades de Aplicación DICOM. Esta verificación se lleva a cabo mediante el establecimiento de una asociación utilizando el servicio *C-ECHO* de *DIMSE-C*.

#### **Comportamiento Entidades**

Una Entidad de Aplicación solicita la verificación de la comunicación a otra entidad remota. Esta solicitud se realiza utilizando la primitiva de petición C-ECHO-RQ. La entidad remota responderá lanzando la primitiva de respuesta C-ECHO-RSP. Cuando la entidad que inició el proceso recibe la respuesta, determina que la verificación se ha completado.

#### **Grupo de Servicio DIMSE-C**

El servicio C-ECHO de DIMSE-C es el mecanismo utilizado para verificar la comunicación entre Entidades de Aplicación pares. El servicio C-ECHO y los parámetros del protocolo se encuentran detallados en la sección 5.11.1.

## Clase SOP de Verificación

La Clase SOP del Servicio de Verificación se compone únicamente del servicio C-ECHO de DIMSE-C, ya que no hay ningún Objeto de Información asociado. El UID de la Clase SOP de Verificación es “1.2.840.10008.1.1”.

## Negociación de la Asociación

El establecimiento de la asociación es la primera fase de cualquier comunicación entre Entidades de Aplicaciones pares. Para la Clase SOP de Verificación, las reglas a seguir son las siguientes:

- ♣ El solicitante de la asociación (SCU) debe utilizar la Sintaxis Abstracta correspondiente al Servicio de Verificación, cuyo UID es idéntico al de la Clase SOP, en la petición A-ASSOCIATE.
- ♣ El que acepta la asociación (SCP) debe aceptar la Sintaxis Abstracta en la respuesta A-ASSOCIATE.

### 5.6.2. Clase de Servicio de Almacenamiento

La *Clase de Servicio de Almacenamiento* define un servicio a nivel de aplicación que facilita la transferencia simple de imágenes entre Entidades de Aplicación pares.

## Comportamiento de las Entidades

La entidad en el papel de SCU invoca el servicio C-STORE de DIMSE-C con una Instancia SOP que cumpla los requisitos del correspondiente Objeto de Información a almacenar. La entidad SCP llevará a cabo el servicio e indicará si se ha llevado a cabo o no con éxito. La entidad SCU deberá recoger el resultado del servicio y tomar las acciones que crea oportunas según el éxito o fracaso del mismo. Los posibles valores que puede tomar la respuesta de la entidad SCP los podemos encontrar en la tabla 5.1.

## Grupo de Servicio DIMSE-C

El servicio C-STORE de DIMSE-C es el mecanismo utilizado para llevar a cabo el Servicio de Almacenamiento. Dicho servicio así como los parámetros del protocolo se encuentran detallados en la sección 5.11.2.

**Tabla 5.1.** Posibles Estados del Servicio C-STORE de DIMSE-C

Estado	Significado	Códigos
Rechazado	Sin Recursos	A7xx
Error	Datos no coinciden con Clase SOP	A9xx
	No Entiendo	Cxxx
Atención	Coerción de Elementos de Datos	B000
	Datos no coinciden con Clase SOP	B007
	Elementos descartados	B006
Éxito		0000

## Negociación de la Asociación

Para la Clase de Servicio de Almacenamiento, la entidad de aplicación en el papel SCU debe incluir en la petición A-ASSOCIATE:

- ♣ Opcionalmente, un Elemento de Negociación Extendida de la Clase SOP, por cada Clase y/o Meta Clase SOP soportada.
- ♣ Una Sintaxis Abstracta por cada Clase y/o Meta Clase SOP soportada.

La entidad de aplicación en el papel SCP deberá aceptar en la respuesta A-ASSOCIATE:

- ♣ Una Sintaxis Abstracta por cada Clase y/o Meta Clase SOP aceptada.
- ♣ Opcionalmente, un Elemento de Negociación Extendida de la Clase SOP, por cada Clase y/o Meta Clase SOP aceptadas.

La Negociación Extendida es opcional. En el caso de que la SCU o la SCP no la soporten, se debe seguir el procedimiento por defecto. Se utiliza para permitir a las Entidades de Aplicación intercambiar información a nivel de conformidad y de opciones soportadas por la implementación.

### Clases SOP soportadas

Las Clases SOP de la Clase de Servicio de Almacenamiento identifican los Objetos de Información Compuestos que se pueden almacenar. En la siguiente tabla podemos ver las Clases SOP Estándar soportadas :

**Tabla 5.2.** Clases SOP Estándar soportadas por el Servicio de Almacenamiento

Nombre Clase SOP	UID Clase SOP
Computed Radiography Image Storage	1.2.840.10008.5.1.4.1.1.1
CT Image Storage	1.2.840.10008.5.1.4.1.1.2
Hardcopy Color Image Storage	1.2.840.10008.5.1.1.30
Hardcopy Grayscale Image Storage	1.2.840.10008.5.1.1.29
MR Image Storage	1.2.840.10008.5.1.4.1.1.4
Nuclear Medicine Image Storage	1.2.840.10008.5.1.4.1.1.20
Positron Emission Image Storage	1.2.840.10008.5.1.4.1.1.128
RT Dose Storage	1.2.840.10008.5.1.4.1.1.481.2
RT Image Storage	1.2.840.10008.5.1.4.1.1.481.1
RT Plan Storage	1.2.840.10008.5.1.4.1.1.481.5
RT Structure Set Storage	1.2.840.10008.5.1.4.1.1.481.3
RT Beams Treatment Record Storage	1.2.840.10008.5.1.4.1.1.481.4
RT Brachy Treatment Record Storage	1.2.840.10008.5.1.4.1.1.481.6
RT Treatment Summary Record Storage	1.2.840.10008.5.1.4.1.1.481.7
Secondary Capture Image Storage	1.2.840.10008.5.1.4.1.1.7
Stand-alone Curve Storage	1.2.840.10008.5.1.4.1.1.9
12-lead ECG Waveform Storage	1.2.840.10008.5.1.4.1.1.9.1.1
General ECG Waveform Storage	1.2.840.10008.5.1.4.1.1.9.1.2
Ambulatory ECG Waveform Storage	1.2.840.10008.5.1.4.1.1.9.1.3
Hemodynamic Waveform Storage	1.2.840.10008.5.1.4.1.1.9.2.1
Cardiac Electrophysiology Waveform Storage	1.2.840.10008.5.1.4.1.1.9.3.1
Basic Voice Audio Waveform Storage	1.2.840.10008.5.1.4.1.1.9.4.1
Stand-alone Modality LUT Storage	1.2.840.10008.5.1.4.1.1.10
Stand-alone Overlay Storage	1.2.840.10008.5.1.4.1.1.8
Stand-alone VOI LUT Storage	1.2.840.10008.5.1.4.1.1.11
Grayscale Softcopy Presentation State Storage	1.2.840.10008.5.1.4.1.1.11.1
Stand-alone PET Curve Storage	1.2.840.10008.5.1.4.1.1.129
Stored Print Storage	1.2.840.10008.5.1.1.27
Ultrasound Image Storage	1.2.840.10008.5.1.4.1.1.6.1
Ultrasound Multi-frame Image Storage	1.2.840.10008.5.1.4.1.1.3.1
X-Ray Angiographic Image Storage	1.2.840.10008.5.1.4.1.1.12.1
X-Ray Radiofluoroscopic Image Storage	1.2.840.10008.5.1.4.1.1.12.2
Digital X-Ray Image Storage - For	1.2.840.10008.5.1.4.1.1.1.1

Presentation	
Digital X-Ray Image Storage - For Processing	1.2.840.10008.5.1.4.1.1.1.1.1
Digital Mammography Image Storage - For Pre-sentation	1.2.840.10008.5.1.4.1.1.1.2
Digital Mammography Image Storage - For Processing	1.2.840.10008.5.1.4.1.1.1.2.1
Digital Intra-oral X-Ray Image Storage - For Presentation	1.2.840.10008.5.1.4.1.1.1.3
Digital Intra-oral X-Ray Image Storage - For Processing	1.2.840.10008.5.1.4.1.1.1.3.1
VL Endoscopic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.1
VL Microscopic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.2
VL Slide-Coordinates Microscopic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.3
VL Photographic Image Storage	1.2.840.10008.5.1.4.1.1.77.1.4
Basic Text SR	1.2.840.10008.5.1.4.1.1.88.11
Enhaced SR	1.2.840.10008.5.1.4.1.1.88.22
Comprehensive SR	1.2.840.10008.5.1.4.1.1.88.33

## 5.7. Codificación

Un Registro de Datos DICOM (*Data Set*) se construye codificando los valores de los Atributos especificados en el correspondiente Objeto de Información. En esta sección vamos a ver cómo se codifican estos Atributos y la estructura general de los Registros de Datos.

### 5.7.1. Juego de caracteres soportados

Los Valores que son texto o cadenas de caracteres se pueden componer tanto de *Caracteres Gráficos* como de *Caracteres de Control*. Según el lenguaje nativo en el que las entidades de aplicación intercambian datos, se utilizará un juego de caracteres u otro. Los juegos de caracteres soportados por DICOM están recogidos en el estándar ISO 8859, a los que hay que añadir los correspondientes al lenguaje Japonés. En la tabla 5.3 podemos ver el Juego de Caracteres Básico utilizado en DICOM.

En DICOM, en las cadenas de texto, una nueva línea se indica con la combinación de los Caracteres de Control “CR+LF”. Esto puede no coincidir con la representación utilizada en algunos sistemas, como por ejemplo los basados en UNIX, que interpretan una nueva línea con el Carácter de Control “LF”. En estos casos, es de

esperar que la implementación convierta a la representación interna correspondiente el formato utilizado en DICOM.

**Tabla 5.3.** Juego de Caracteres Básico DICOM

	00	01	02	03	04	05	06	07
00			SP	0	@	P	'	p
01			!	1	A	Q	a	q
02			"	2	B	R	b	r
03			#	3	C	S	c	s
04			\$	4	D	T	d	t
05			%	5	E	U	e	u
06			&	6	F	V	f	v
07			'	7	G	W	g	w
08			(	8	H	X	h	x
09			)	9	I	Y	i	y
10	LF		*	:	J	Z	j	z
11		ESC	+	;	K	[	k	{
12	FF		,	<	L	\	l	
13	CR		-	=	M	]	m	}
14			.	>	N	^	n	~
15			/	?	O	_	o	

### 5.7.2. Representación de Valores

La *Representación del Valor*, en adelante VR (*Value Representation*), describe el tipo de dato y el formato a utilizar para codificar el valor de un determinado Atributo. En la tabla 5.4 podemos ver algunas de las representaciones utilizadas en DICOM.

**Tabla 5.4.** Algunas Representaciones de Valor definidas en DICOM

VR Nombre	Definición	Repertorio Caracteres	Longitud
AS Edad	Cadena de caracteres con uno de los siguientes formatos: nnnD, nnnW, nnnM, nnnY, donde nnn contiene el número de días si es D, semanas para W, meses para M o años para Y.	"0"- "9", "D", "W", "M", "Y" del Juego de Caracteres Básico	4 bytes fijo
CS Cadena Caracteres	Cadena de caracteres con espacios (código 20H) no significativos por delante o por detrás.	Caracteres en Mayúscula, "0"- "9", espacios (SP) y guión bajo "_" del Juego de Caracteres Básico.	16 bytes máximo
DA	Cadena de caracteres con el formato	"0"- "9" del	8 bytes

Fecha	yyyymmdd, donde yyyy contiene el año, mm el mes y dd el día.	Juego de Caracteres Básico.	fijo
FL Punto flotante simple	Número binario en punto flotante de precisión simple representado en formato IEEE 754:1985 32-bit.	no aplicable.	4 bytes fijo
FL Punto flotante doble	Número binario en punto flotante de precisión doble representado en formato IEEE 754:1985 64-bit.	no aplicable.	8 bytes fijo
OB Cadena de Bytes	Cadena de bytes donde la codificación del contenido se especifica en la Sintaxis de Transferencia negociada. Es insensible a la ordenación de bytes utilizada (Little/Big Endian).	no aplicable.	Ver Sintaxis de Transferencia negociada
OW Cadena de Palabras	Cadena de palabras de 16-bits donde la codificación del contenido se especifica en la Sintaxis de Transferencia negociada. Requiere el intercambio de bytes dentro de cada palabra cuando se pasa de ordenación Little Endian a Big Endian y viceversa.	no aplicable.	Ver Sintaxis de Transferencia negociada
PN Nombre Persona	Cadena de caracteres codificada utilizando un formato de 5 componentes. Los cinco componentes, en orden de aparición son los siguientes: apellidos, nombre, nombre pila, prefijo y sufijo. El carácter delimitador es el “^” correspondiente al código 5EH. El delimitador debe usarse para separar los componentes, aunque sean nulos. Los campos nulos en la cola pueden eliminarse junto con el separador correspondiente.	Juego de Caracteres Básico, excluyendo los caracteres de control LF, FF y CR, pero permitiendo el ESC	64 caracteres máximo
SQ Secuencia de Ítems	El valor es una secuencia de cero o más Ítems.	no aplicable.	no aplicable.

### 5.7.3. Multiplicidad y Delimitación

La *Multiplicidad del Valor* de un Elemento de Dato especifica el número de valores que pueden ser codificados en dicho campo. Cuando hay múltiples valores, éstos han de delimitarse de la siguiente forma:

- ♣ Para cadenas de caracteres, el carácter 5CH, es decir, la barra invertida “\”, debe utilizarse como delimitador entre los distintos valores.
- ♣ Para valores binarios de longitud fija, se deben codificar como una serie de valores seguidos, sin delimitador alguno.

#### 5.7.4. El Registro de Datos DICOM

Un *Registro de Datos DICOM (Data Set)* representa una instancia de un Objeto de Información. Se construye con *Elementos de Datos*, los cuales no son más que la codificación de los Atributos de ese Objeto de Información.

##### Elementos de Datos

Un Elemento de Datos se identifica de forma única mediante la *Etiqueta*. Dentro de un Registro, los Elementos de Datos deben estar ordenados de forma creciente según esta Etiqueta.

Un Elemento de Datos puede tener tres estructuras. Dos de ellas contienen la VR utilizada en la codificación (*VR explícita*), pero difieren en la forma de expresar la Longitud. La otra no contiene la VR (*VR implícita*). Además todas ellas han de llevar la Etiqueta, Longitud y el Valor en sí del Elemento de Datos (ver figura 5.5).

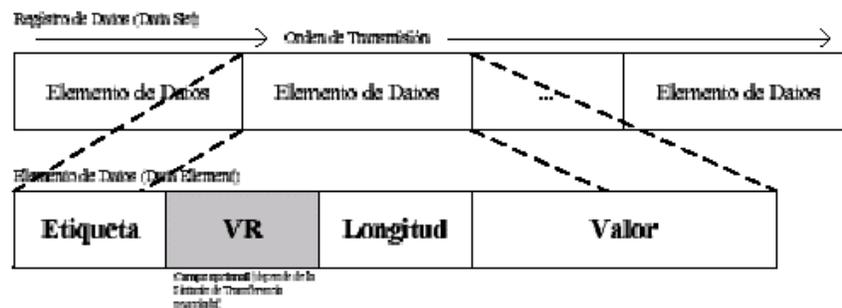


Figura 5.5. El Registro de Datos DICOM

Los campos de que se compone son:

- ♣ *Etiqueta*: par ordenado de enteros sin signo de 16 bits que representan el número de grupo, seguido del número de elemento, en el formato (gggg,eeee).
- ♣ *VR*: cadena de caracteres de 2 bytes que contienen la VR del Elemento. La VR de cada Elemento de Datos viene especificada en la Parte 6 del Estándar DICOM, titulada “Diccionario de Datos”.
- ♣ *Longitud*: puede ser de dos tipos:

- *Explícita*: entero sin signo de 16 o 32 bits (según la estructura utilizada) que contiene el número de bytes que ocupa el campo Valor.
  - *Indefinida*: entero de 32 bits con el valor ‘FFFFFFFFH’. Se utiliza para Elementos de Datos que tiene una VR correspondiente a ‘Secuencia de Items’ (SQ) o ‘Desconocido’ (UN). Para VRs ‘OW’ o ‘OB’ se puede utilizar según la Sintaxis de Transferencia negociada.
- ♣ *Valor*: un número par de bytes que contienen el valor del atributo codificado. El tipo de dato almacenado en este campo vendrá determinado por la VR utilizada. Si la multiplicidad del Valor es mayor que 1, se pueden codificar múltiples valores dentro de este campo, utilizando los delimitadores vistos en la sección 5.7.3.

### **Ordenación de bytes**

Una cuestión muy importante en la codificación de un Registro de Datos es la ordenación de los bytes, que debe ser negociada en el establecimiento de la Asociación. Por defecto, la Sintaxis de Transferencia utilizada en DICOM, y que debe ser soportada por todas las implementaciones DICOM, utiliza la ordenación ‘Little Endian’. Cuando la ordenación de los bytes utilizada no coincida con la ordenación interna de la máquina, habrá que proceder al intercambio de bytes. Los Registros de Datos afectados serán aquellos que tienen VR con valores que ocupan múltiples bytes que no sean cadenas de caracteres, ya que éstas no son más que una cadena de bytes individuales, por lo que no necesitan ningún intercambio.

### **Tipo de los Atributos**

Un Atributo, codificado como un Elemento de Dato, puede o no ser requerido en un Registro de Datos. Esto viene determinado por el Tipo de Atributo. El Tipo del Atributo especifica si el atributo es obligatorio, condicional u opcional. Los tipos posibles son:

- ♣ *Tipo 1, Obligatorio No Nulo*: el campo valor debe contener datos válidos según la VR y la multiplicidad, y el campo longitud no debe ser cero.

- ♣ *Tipo 1C, Condicional Tipo 1:* según determinadas circunstancias, especificadas en el Objeto de Información, estos atributos pasan a tener los mismos requerimientos que los de “Tipo 1”. Si no se dan esas circunstancias, no deben codificarse.
- ♣ *Tipo 2, Obligatorio:* el atributo debe codificarse, pero a diferencia de los de “Tipo 1”, se permite un valor nulo, es decir, longitud cero y ningún valor.
- ♣ *Tipo 2C, Condicional Tipo 2:* al igual que los de “Tipo 1C”, si se dan determinadas condiciones, pasan a considerarse como de “Tipo 2”.
- ♣ *Tipo 3, Opcional:* el Atributo puede aparecer o no, según la implementación. También pueden tener un valor nulo.

### **Encadenamiento de Elementos de Datos**

Hay una VR especial, llamada “*Secuencia de Items*” o “SQ”, la cual consiste en una secuencia de cero o más *Items*, donde cada uno de ellos contiene un conjunto de Elementos de Datos. Esta VR permite una forma fácil de codificar estructuras simples de Elementos de Datos que se repiten, Objetos de Información más complejos, e incluso permiten el anidamiento en múltiples niveles.

#### **5.7.5. Codificación de Imágenes**

Para el intercambio de información gráfica disponemos de los Elementos de Datos básicos “*Pixel Data Element*” y “*Overlay Data Element*”. Con ellos y junto con otros Elementos de Datos adicionales se describe la forma en que se ha codificado la información gráfica de las imágenes y transparencias, y cómo ha de interpretarse.

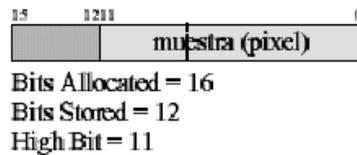
### **Codificación de la información gráfica**

Hay tres Elementos de Datos que definen la estructura de la información gráfica de una imagen:

- ♣ *Bits Allocated:* indica el número de bits reservados para cada píxel de la imagen.
- ♣ *Bits Stored:* indica cuantos bits se utilizan realmente para codificar el píxel.

- ♣ *High Bit*: indica donde se encuentra situado el bit de mayor importancia respecto de los bits reservados.

El valor del píxel codificado debe ser un número binario entero con signo, en complemento a 2, o un número binario entero sin signo. En el caso de enteros con signo, el bit correspondiente al signo debe ser el especificado en High Bit. Los valores mínimos y máximos codificables se especifican respectivamente en los Elementos de Datos “*Smallest Image Pixel Value*” y “*Largest Image Pixel Value*”.



**Figura 5.6.** Ejemplo de codificación de un píxel

### Formatos de Codificación

La información gráfica de una imagen puede ir codificada en dos formatos bien distintos:

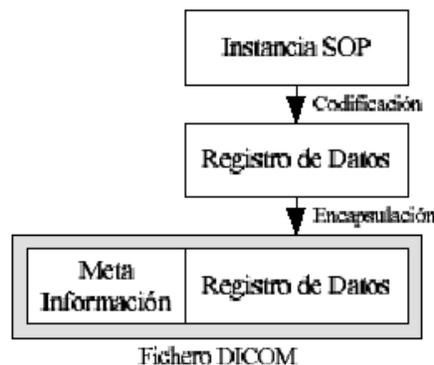
- ♣ *Nativo*: sin comprimir. Los distintos píxeles de la imagen se codifican como la concatenación directa de los bits de cada píxel, uno detrás de otro. Normalmente se utiliza la VR “Cadena de Palabras” (OW), a menos que el número de bits por píxel no sea superior a 8, en cuyo caso puede utilizarse la VR “Cadena de Bytes” (OB).
- ♣ *Encapsulado*: comprimido. Se utiliza un determinado formato de compresión para codificar los distintos píxeles de la imagen. La especificación de dicho formato está fuera del alcance del Estándar DICOM, pero debe estar recogido en alguna de las Sintaxis de Transferencia definidas. En este caso, se utiliza siempre la VR correspondiente “Cadena de Bytes” (OB).

En DICOM se recogen tres formatos básicos de compresión:

- ♣ *JPEG*: formato de compresión recogido en el Estándar Internacional ISO/IS-10918-1 y 2 para la compresión y codificación de imágenes fijas. Define dos procesos de codificación, con pérdidas (basado en la DCT o Transformada del Coseno) y sin pérdidas (basado en DPCM o Modulación por Pulsos Diferenciales), aunque normalmente se utiliza el primero, ya que proporciona mucha más compresión.
- ♣ *JPEG-LS*: formato de compresión recogido en el Estándar Internacional ISO/IS-14495-1 para compresión y codificación de imágenes fijas. Proporciona un método de compresión casi sin pérdidas, que puede llegar a ser sin pérdidas si se minimiza el error absoluto a cero durante la compresión. Se basa en técnicas muy distintas respecto al formato JPEG sin pérdidas, y consigue una mayor compresión a la vez que simplifica bastante el proceso.
- ♣ *RLE (Run Length Encoding)*: es un formato de compresión sin pérdidas muy simple, consistente en codificar valores idénticos consecutivos en sólo de 2 bytes, reduciendo así el tamaño final de la imagen.

## 5.8. El Formato de Fichero DICOM

El Formato de Fichero DICOM permite encapsular en un fichero el Registro de Datos correspondiente a una Instancia SOP. Como podemos ver en la figura 5.7, el Registro de Datos se coloca en el fichero justo detrás de la Metainformación de Fichero. Cada fichero DICOM contiene una y sólo una Instancia SOP.



**Figura 5.7.** Formato de fichero DICOM

### 5.8.1. Metainformación de Fichero DICOM

La *Metainformación* contiene la información necesaria para identificar al Registro de Datos encapsulado en el fichero. Esta cabecera consta de 128 bytes de *Preámbulo*, 4 bytes de prefijo DICOM y una serie de *Elementos de Metainformación de Fichero*. Esta cabecera debe estar presente en todos los ficheros DICOM. El Preámbulo está disponible para el uso específico de las implementaciones, por lo que no requiere ningún tipo de estructura. La intención de este preámbulo es facilitar el acceso a las imágenes y a otra información del fichero DICOM. Si no se utiliza, debe estar relleno de ceros. A continuación del Preámbulo, viene el prefijo DICOM, el cual consiste en 4 bytes conteniendo los caracteres ‘DICM’, codificados como caracteres en mayúsculas del Juego de Caracteres Básico. La estructura completa la tenemos en la tabla 5.5.

Excepto para los 128 bytes del Preámbulo y los 4 bytes del Prefijo DICOM, la cabecera de Metainformación debe codificarse utilizando la Sintaxis de Transferencia ‘Explícita Little Endian’.

**Tabla 5.5.** Estructura de Fichero DICOM

<b>Campo</b>	<b>Etiqueta</b>	<b>Tipo</b>	<b>Descripción</b>
Preámbulo	—	1	Campo Fijo de 128 bytes para uso específico de las aplicaciones. Si no se utiliza, debe rellenarse con ceros.
Prefijo DICOM	—	1	Cuatro bytes conteniendo la cadena de caracteres “DICM”.
Longitud Grupo	(0002,0000)	1	Número de bytes que siguen a este Elemento de Metainformación.
Versión Metainformación	(0002,0001)	1	Campo de 2 bytes que identifica la versión de esta cabecera de Metainformación. Debe tener el valor “01H”, correspondiente a esta versión del Estándar.
UID Clase SOP	(0002,0002)	1	Identifica la Clase SOP asociada con el Registro de Datos.
UID Instancia SOP	(0002,0003)	1	Identifica la Instancia SOP del Registro de Datos encapsulado en el fichero.
UID Sintaxis Transferencia	(0002,0010)	1	Identifica la Sintaxis de Transferencia utilizada para codificar el Registro de Datos. No se aplica a la Metainformación

UID Implementación	(0002,0012)	1	Identifica la implementación que escribió este fichero y su contenido
Versión Implementación	(0002,0013)	3	Identifica una determinada versión dentro de un UID de Implementación.
Título Entidad Aplicación	(0002,0016)	3	Título de la Entidad de Aplicación que escribió este fichero y su contenido o qué lo actualizó.
UID Creador Información Privada	(0002,0100)	3	El UID del creador de la información privada que se codifica en el siguiente campo.
Información Privada	(0002,0102)	3	Contiene información privada. Si se utiliza, el creador se debe identificar utilizando el campo anterior.

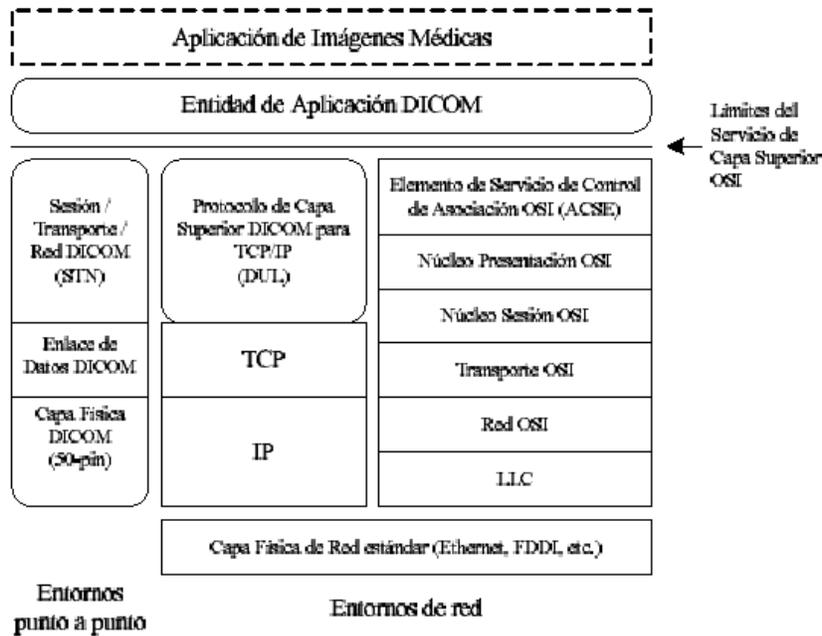
### 5.8.2. Encapsulación del Registro de Datos

Cada fichero DICOM debe contener un único Registro de Datos representando una Instancia SOP. Esto no quiere decir que haya solo una imagen en el fichero, ya que según el Objeto de Información, una Instancia SOP puede contener varios *frames* dentro de una imagen. La Sintaxis de Transferencia utilizada para codificar la Instancia SOP será la especificada en el campo correspondiente en la cabecera de Metainformación del fichero.

## 5.9. Intercambio de Mensajes

### 5.9.1. DICOM y el Modelo de Referencia OSI

El modelo utilizado para la interconexión de equipos médicos siguiendo el Estándar DICOM es el Modelo de Referencia OSI, compuesto de siete capas. DICOM utiliza el Servicio OSI de Capa Superior para separar el intercambio de mensajes DICOM a nivel de Aplicación del soporte de comunicación proporcionado por las capas inferiores. La interfaz de este Servicio permite a las Entidades de Aplicación establecer asociaciones, transferir mensajes y terminar asociaciones. En la figura 5.8 podemos ver las torres de protocolos disponibles en DICOM.



**Figura 5.8.** DICOM y el Modelo de Referencia OSI

Hay tres opciones a la hora de interconectar equipos mediante protocolos DICOM:

- ♣ Utilizar una pila de protocolos OSI mínima, con Núcleo de Sesión OSI, Núcleo de Presentación OSI y ACSE. Con esto reducimos la sobrecarga de capas a la vez que mantenemos total conformidad con el Estándar OSI.
- ♣ Utilizar un protocolo de Capa Superior que expanda TCP/IP. Combina los protocolos de capa superior OSI en una sola capa que proporciona los mismos servicios (*Capa DUL, DICOM Upper Layer*).
- ♣ Utilizar una pila de protocolos punto a punto compatible DICOM. Las Entidades de Aplicación DICOM utilizan los servicios de datos de Asociación y Presentación del Servicio de Capa Superior OSI (ver sección 5.12).

El *Elemento de Servicio de Control de Asociación* o ACSE expande el Servicio de Capa de Presentación con las capacidades de establecimiento y terminación de asociaciones. En el caso de TCP/IP, el equivalente completo de ACSE lo proporciona el Servicio de Capa Superior DICOM (Capa DUL), mientras que en el caso de

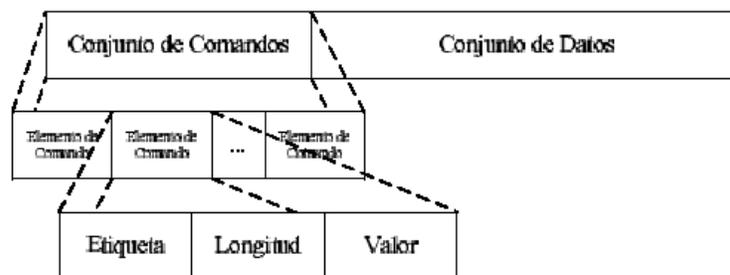
comunicaciones punto a punto, un subconjunto mínimo de ACSE se proporciona en las capas de Sesión, Transporte y Red.

Las Entidades de Aplicación DICOM utilizarán los servicios proporcionados por el *Elemento de Servicio de Mensajes DICOM* o DIMSE, el cual especifica dos tipos de servicios:

- ♣ DIMSE-C: soporta operaciones asociadas con Clases SOP Compuestas, además de proporcionar cierta compatibilidad con versiones anteriores del Estándar DICOM.
- ♣ DIMSE-N: soporta operaciones asociadas con Clases SOP Normalizadas y proporciona un conjunto bastante amplio de operaciones y notificaciones relacionadas con objetos.

### 5.9.2. Estructura de los Mensajes DICOM

La información se intercambia entre los equipos mediante los Mensajes DICOM. Un Mensaje DICOM se compone de un *Conjunto de Comandos* además de un *Conjunto de Datos* opcional. El Conjunto de Comandos se utiliza para indicar las operaciones o notificaciones que se deben realizar con el Conjunto de Datos que acompaña (ver figura 5.9).



**Figura 5.9.** Estructura de un Mensaje DICOM

El Conjunto de Comandos se compone de Elementos de Comandos, los cuales contienen los valores codificados para cada campo del Conjunto de Comando según la

semántica especificada en el protocolo DIMSE. La codificación del Conjunto de Comandos debe hacerse utilizando siempre la ordenación Little Endian.

Cada Elemento de Comando se compone de tres campos:

- ♣ *Etiqueta*: par ordenado de enteros sin signo de 16 bits representando el Número de Grupo seguido del Número de Elemento, en el formato (gggg,eeee).
- ♣ *Longitud*: entero sin signo de 32 bits representando la Longitud Explícita que ocupa el Valor. Debe ser siempre un valor par y no incluye la Etiqueta ni el propio campo Longitud.
- ♣ *Valor*: un número par de bytes que contienen uno o más Valores. El tipo de valor codificado en este campo vendrá dado por la VR del Elemento de Comando. Si la Multiplicidad del Valor es mayor que 1, se utilizarán delimitadores, como los utilizados en la sección 5.7.3.

## 5.10. DIMSE

El Elemento de Servicio de Mensajes DICOM, conocido como DIMSE, soporta la comunicación entre Usuarios del Servicio DIMSE pares. Un usuario DIMSE lleva a cabo uno de los dos papeles posibles:

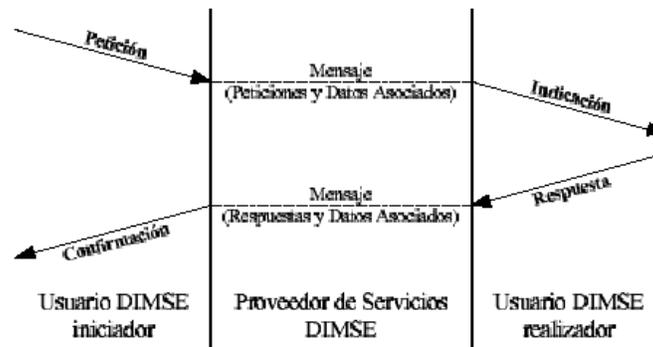
- ♣ Usuario DIMSE iniciador
- ♣ Usuario DIMSE realizador

Los Usuarios del Servicio DIMSE hacen uso de las primitivas proporcionadas por el proveedor de servicios DIMSE. Las primitivas pueden ser:

- ♣ de petición
- ♣ de indicación
- ♣ de respuesta
- ♣ de confirmación

Para completar con éxito un servicio DIMSE, las primitivas deben ser utilizadas de la siguiente forma:

- ♣ El usuario iniciador lanza una primitiva de petición al proveedor de servicios DIMSE.
- ♣ El proveedor de servicios recibe la petición y lanza una primitiva de indicación al usuario DIMSE realizador.
- ♣ El usuario realizador recibe la primitiva de indicación del proveedor de servicios y realiza las operaciones oportunas.
- ♣ El usuario realizador lanza una primitiva de respuesta al proveedor de servicios DIMSE.
- ♣ El proveedor de servicios recibe la primitiva de respuesta del usuario realizador y lanza una primitiva de confirmación hacia el usuario iniciador.
- ♣ El usuario DIMSE iniciador recibe la primitiva de confirmación del proveedor de servicios, con lo que completa el servicio DIMSE.



**Figura 5.10.** Primitivas de Servicio DIMSE

DIMSE proporciona dos tipos de servicio de transferencia de información que pueden utilizar las Entidades de Aplicación:

- ♣ *Servicio de Notificación:* permite a una Entidad de Aplicación DICOM notificar a otra sobre la ocurrencia de algún evento o cambio de estado.
- ♣ *Servicio de Operación:* permite a una Entidad de Aplicación DICOM requerir explícitamente que se realice una operación sobre una Instancia SOP gestionada por otra Entidad de Aplicación.

Ambos tipos de servicios, en el contexto de una Asociación, pueden usarse de dos modos: síncrono y asíncrono. En el modo síncrono, el Usuario DIMSE iniciador, una vez establecida la Asociación, espera la respuesta del Usuario DIMSE realizador antes de lanzar otra petición. En el modo asíncrono, el Usuario DIMSE iniciador puede continuar lanzando peticiones sin tener que esperar a que vayan llegando las correspondientes respuestas

La selección del modo de trabajo se realiza durante el establecimiento de la Asociación. Por defecto, el modo de trabajo es síncrono, por lo que debe ser soportado por todas las implementaciones DICOM. El modo asíncrono es opcional, y el número máximo de peticiones que se pueden mantener en espera de respuesta se negocia durante la fase de establecimiento de la Asociación.

### 5.10.1. Servicios DIMSE

Hay dos grupos de servicios DIMSE, según el tipo de instancia SOP a los que se aplique:

- ♣ *DIMSE-N*: se aplican a Instancias SOP Normalizadas.
- ♣ *DIMSE-C*: se aplican a Instancias SOP Compuestas

En la tabla 5.6 podemos ver todos los servicios DIMSE definidos en la versión actual del Estándar DICOM.

**Tabla 5.6.** Servicios DIMSE

Nombre	Grupo	Tipo
C-STORE	DIMSE-C	operación
C-GET	...	operación
C-MOVE	...	operación
C-FIND	...	operación
C-ECHO	...	operación
N-EVENT-REPORT	DIMSE-N	notificación
N-GET	...	operación
N-SET	...	operación
N-ACTION	...	operación
N-CREATE	...	operación
N-DELETE	...	operación

Dentro de los servicios del grupo DIMSE-C sólo disponemos de servicios del tipo operación. La función de cada uno de ellos es la siguiente:

- ♣ C-STORE: para que el usuario DIMSE realizador almacene una Instancia SOP Compuesta.
- ♣ C-GET: para requerir información acerca de una o más Instancias SOP Compuestas al usuario DIMSE realizador, basándose en los atributos proporcionados por el usuario DIMSE iniciador.
- ♣ C-MOVE: para pedir el traslado de la información de una o más Instancias SOP Compuestas desde el usuario DIMSE realizador a un tercer usuario DIMSE, basándose en los atributos proporcionados.
- ♣ C-FIND: para comparar una serie de atributos con las Instancias SOP gestionadas en el usuario DIMSE realizador, que devolverá una lista con los atributos pedidos y sus correspondientes valores.
- ♣ C-ECHO: para verificar la comunicación extremo a extremo entre los usuarios DIMSE.

En el grupo de servicios DIMSE-N hay servicios tanto de notificación como de operación. La función de cada uno de ellos es la siguiente:

- ♣ N-EVENT-REPORT: para informar al usuario DIMSE realizador acerca de algún evento relacionado con una Instancia SOP Normalizada.
- ♣ N-GET: para requerir información al usuario DIMSE realizador.
- ♣ N-SET: para pedir la modificación de información al usuario DIMSE realizador.
- ♣ N-ACTION: para pedir al usuario DIMSE realizador que lleve a cabo alguna acción.
- ♣ N-CREATE: para pedir la creación de una Instancia SOP.
- ♣ N-DELETE: para pedir la destrucción de una Instancia SOP.

### 5.10.2. Procedimientos DIMSE

Todas las operaciones y notificaciones DIMSE son servicios confirmados, por lo que el Usuario DIMSE realizador debe dar la respuesta de cada una de ellas en la misma asociación en la que se recibió la petición. Cada servicio DIMSE se realiza mediante el

uso de una o más primitivas. El cómo deben reaccionar los usuarios a estas primitivas está recogido en los procedimientos definidos para cada Servicio.

### **Suboperaciones**

Algunos servicios DIMSE son atómicos, en el sentido de que sólo se necesita una operación o notificación para llevarlos a cabo. Otros, sin embargo, requieren el uso de una o más suboperaciones, que serán realizadas mediante las correspondientes primitivas.

### **Múltiples Respuestas**

Los servicios DIMSE requieren de una o más primitivas de respuesta como resultado de la invocación de un servicio. Según la información incluida en la primitiva de petición, se deberán devolver o no múltiples respuestas.

### **Cancelación**

Algunos servicios DIMSE permiten la cancelación del servicio mediante alguna primitiva. Esto permite al usuario DIMSE iniciador solicitar la terminación del servicio después de haber cursado la petición del mismo, pero antes de que se haya completado.

En la tabla 5.7 podemos ver qué servicios DIMSE admiten los distintos procedimientos que acabamos de ver.

## **5.11. DIMSE-C**

A continuación vamos a ver con detalle algunos de los servicios DIMSE utilizados en este Proyecto, como son C-STORE y C-ECHO.

### 5.11.1. Servicio C-ECHO

El servicio C-ECHO lo utilizan los usuarios DIMSE para verificar las comunicaciones extremo a extremo.

**Tabla 5.7.** Procedimientos permitidos para los Servicios DIMSE

Nombre	Sub-operaciones	Múltiples Respuestas	Cancelación
C-STORE	—	—	—
C-GET	Sí	Sí	Sí
C-MOVE	Sí	Sí	Sí
C-FIND	—	Sí	Sí
C-ECHO	—	—	—
N-EVENT-REPORT	—	—	—
N-GET	—	—	—
N-SET	—	—	—
N-ACTION	—	—	—
N-CREATE	—	—	—
N-DELETE	—	—	—

### Parámetros

Los parámetros utilizados en el Servicio C-ECHO los podemos ver en la tabla 5.8.

**Tabla 5.8.** Parámetros del Servicio C-ECHO

Nombre Parámetro	Pet / Ind	Rsp / Conf
ID Mensaje	M	—
ID Mensaje Respondido	—	M
UID Clase SOP	M	U (=)
Estado	—	M

- ♣ *ID Mensaje*: este parámetro identifica la operación. Se utiliza para distinguir esta operación de otras notificaciones u operaciones que pueda estar llevando a cabo el Proveedor del Servicio DIMSE en este momento, por tanto no debe haber operaciones en progreso con un mismo ID de Mensaje.
- ♣ *ID Mensaje Respondido*: especifica el ID del Mensaje de la petición / indicación de la operación al cual la respuesta / confirmación se aplica.

- ♣ *UID Clase SOP*: para la petición / indicación, este parámetro especifica la Clase SOP que se quiere almacenar. Se puede poner en la respuesta / confirmación, con el mismo valor que en la petición / indicación.
- ♣ *Estado*: contiene el estado de la respuesta. Debe tener siempre un valor de éxito.

## Procedimientos

Los procedimientos aplicables al Servicio C-ECHO son los siguientes:

1. El usuario DIMSE iniciador pide la verificación de las comunicaciones al usuario DIMSE realizador mediante el uso de la primitiva de petición C-ECHO al Proveedor de Servicios DIMSE.
2. El Proveedor de Servicios lanza una primitiva de indicación al usuario realizador.
3. El usuario realizador verifica la comunicación lanzando una primitiva de respuesta C-ECHO al Proveedor de Servicios DIMSE.
4. El Proveedor de Servicios lanza una primitiva de confirmación C-ECHO al usuario iniciador, completando el Servicio C-ECHO.

## Protocolo

La información necesaria para las primitivas de petición e indicación se recoge en el mensaje C-ECHO-RQ, mientras que para las de respuestas y confirmación tenemos el mensaje C-ECHO-RSP. En las siguientes tablas (tablas 5.9 y 5.10) podemos ver los distintos campos que componen estos mensajes.

Los procedimientos del Servicio C-ECHO se inician cuando el usuario DIMSE iniciador lanza una primitiva de petición C-ECHO. La máquina de protocolo DIMSE-C, al recibir esta primitiva deberá:

- ♣ Construir un mensaje del tipo C-ECHO-RQ.
- ♣ Mandar el mensaje utilizando el servicio P-DATA.

Una vez recibido el mensaje C-ECHO-RQ, la máquina de protocolo DIMSE-C deberá lanzar una primitiva de indicación al usuario DIMSE realizador.

**Tabla 5.9.** Campos del Mensaje C-ECHO-RQ

Campo	Etiqueta	VR	VM	Descripción
Longitud Grupo	(0000,0000)	UL	1	Número par de bytes desde el final del campo Valor hasta el comienzo del siguiente grupo.
UID Clase SOP	(0000,0002)	UI	1	UID de la Clase SOP de Verificación.
Comando	(0000,0100)	US	1	Debe ser "0030H" para el mensaje C-ECHO-RQ.
ID Mensaje	(0000,0110)	US	1	Distingue un mensaje de otros mensajes.
Tipo Registro de Datos	(0000,0800)	US	1	Indica que hay un Registro de Datos presente en el mensaje. Debe ser cualquier valor distinto de "0101H" (Nulo).

**Tabla 5.10.** Campos del Mensaje C-ECHO-RSP

Campo	Etiqueta	VR	VM	Descripción
Longitud Grupo	(0000,0000)	UL	1	Número par de bytes desde el final del campo Valor hasta el comienzo del siguiente grupo.
UID Clase SOP	(0000,0002)	UI	1	UID de la Clase SOP de Verificación.
Comando	(0000,0100)	US	1	Debe ser "0030H" para el mensaje C-ECHO-RSP.
ID Mensaje Respondido	(0000,0120)	US	1	Debe ser el valor del campo "ID del Mensaje" del mensaje C-ECHO-RQ.
Tipo Registro de Datos	(0000,0800)	US	1	Debe contener el valor "0101H" (Nulo), indicando que no hay Registro de Datos presente.
Estado	(0000,0900)	US	1	Indica el Estado de la respuesta. Debe ser siempre de "Éxito".

Cuando la máquina de protocolo DIMSE-C reciba la primitiva de respuesta, lanzada por el Usuario DIMSE realizador, deberá:

- ♣ Construir un mensaje del tipo C-ECHO-RSP.
- ♣ Mandar el mensaje utilizando el servicio P-DATA.

Una vez recibido el mensaje C-ECHO-RSP, la máquina de protocolo DIMSE-C deberá lanzar una primitiva de confirmación al usuario DIMSE iniciador.

### 5.11.2. Servicio C-STORE

El Servicio C-STORE lo utiliza un Usuario DIMSE para almacenar una Instancia SOP Compuesta en otro Usuario DIMSE par.

#### Parámetros

Los parámetros utilizados en este servicio los encontramos en la tabla 5.11.

**Tabla 5.11.** Parámetros del Servicio C-STORE

Nombre Parámetro	Pet / Ind	Rsp / Conf
ID Mensaje	M	—
ID Mensaje Respondido	—	M
UID Clase SOP	M	U (=)
UID Instancia SOP	M	U (=)
Prioridad	M	—
Título Entidad Aplicación C-MOVE	U	—
ID Mensaje C-MOVE	U	—
Registro de Datos	M	—
Estado	—	M

A continuación vamos a ver los parámetros con más detenimiento.

- ♣ *ID Mensaje*: este parámetro identifica la operación. Se utiliza para distinguir esta operación de otras notificaciones u operaciones que pueda estar llevando a cabo el Proveedor del Servicio DIMSE en este momento. Por tanto, no debe haber varias operaciones en progreso con un mismo ID de Mensaje.
- ♣ *ID Mensaje Respondido*: especifica el ID del Mensaje de petición / indicación al que se está respondiendo.
- ♣ *UID Clase SOP*: para la petición / indicación, este parámetro especifica la Clase SOP de la instancia que se quiere almacenar. Se puede poner en la respuesta / confirmación, con el mismo valor que en la petición / indicación.

- ♣ *UID Instancia SOP*: para la petición / indicación, este parámetro especifica la Instancia SOP a almacenar. Se puede poner en la respuesta / confirmación, con el mismo valor que en la petición / indicación.
- ♣ *Prioridad*: especifica la prioridad de la operación C-STORE. Puede ser LOW (baja), MEDIUM (media) o HIGH (alta).
- ♣ *Título Entidad Aplicación C-MOVE*: especifica el Título de la Entidad de Aplicación que invocó la petición C-MOVE que ha originado esta suboperación C-STORE.
- ♣ *ID Mensaje C-MOVE*: especifica el ID del Mensaje de la petición / indicación de la operación original C-MOVE que ha originado esta suboperación C-STORE.
- ♣ *Registro de Datos*: contiene los Atributos de la Instancia SOP que se quiere almacenar.
- ♣ *Estado*: contiene la notificación de error o éxito de la operación. Debe ser colocado por el usuario DIMSE realizador en la respuesta / confirmación. Los posibles valores son:

- *Rechazado: Sin Recursos*; indica que el usuario DIMSE realizador no ha podido almacenar la Instancia SOP debido a que no dispone de recursos.
- *Rechazado: Clase SOP No Soportada*; indica que el usuario DIMSE realizador no ha podido almacenar la instancia debido a que no soporta la Clase SOP a la que pertenece.
- *Error: Desconocido*; indica que el usuario DIMSE realizador no ha podido almacenar la Instancia SOP debido a que no entiende algunos Elementos de Datos.
- *Error: Conjunto de Datos no coincide con Clase*; indica que el usuario DIMSE realizador no ha podido almacenar la Instancia SOP debido a que ésta no coincide con la Clase SOP especificada.
- *Aviso*; indica que el usuario DIMSE realizador pudo almacenar la Instancia SOP, pero detectó un probable error.
- *Éxito*; la operación se llevó a cabo con éxito.

## Procedimientos

Los procedimientos aplicables al Servicio C-STORE son los siguientes:

1. El usuario DIMSE iniciador pide al usuario DIMSE realizador que almacene una Instancia SOP Compuesta mediante el lanzamiento de una primitiva de petición al Proveedor de Servicios DIMSE.
2. El Proveedor de Servicios lanza una primitiva de indicación C-STORE al usuario realizador.
3. El usuario realizador informa de la aceptación o rechazo de la primitiva de petición C-STORE mediante el lanzamiento de la primitiva de respuesta C-STORE al Proveedor de Servicios.
4. El Proveedor de Servicios lanza una primitiva de confirmación CSTORE al usuario DIMSE iniciador, completando la operación CSTORE.

El usuario DIMSE realizador puede devolver una primitiva de respuesta C-STORE con el estado de ‘Fallo’ o ‘Rechazo’ antes de que la primitiva de indicación C-STORE haya sido completamente transmitida por el usuario iniciador. Una respuesta C-STORE con estado de ‘Éxito’ o ‘Aviso’ no puede ser devuelta antes de que haya llegado completamente la primitiva de indicación.

## Protocolo

La información necesaria para las primitivas de petición e indicación se recoge en el mensaje C-STORE-RQ, mientras que para las de respuestas y confirmación tenemos el mensaje C-STORE-RSP. En las siguientes tablas (tablas 5.12 y 5.13) podemos ver los distintos campos de que se componen estos mensajes.

Los procedimientos del Servicio C-STORE se inician cuando el usuario DIMSE iniciador lanza una primitiva de petición C-STORE. La máquina de protocolo DIMSE-C, al recibir esta primitiva deberá:

- ♣ Construir un mensaje del tipo C-STORE-RQ.
- ♣ Mandar el mensaje utilizando el servicio P-DATA.

**Tabla 5.12.** Campos del Mensaje C-STORE-RQ

<b>Campo</b>	<b>Etiqueta</b>	<b>VR</b>	<b>VM</b>	<b>Descripción</b>
Longitud Grupo	(0000,0000)	UL	1	Número par de bytes desde el final del campo Valor hasta el comienzo del siguiente grupo.
UID Clase SOP	(0000,0002)	UI	1	UID de la Clase SOP a la que pertenece la Instancia SOP que se quiere almacenar
Comando	(0000,0100)	US	1	Debe ser "0001H" para el mensaje C-STORE-RQ
ID Mensaje	(0000,0110)	US	1	Distingue un mensaje de otros mensajes
Prioridad	(0000,0700)	US	1	La prioridad debe ser uno de los siguientes valores: LOW = "0002H" MEDIUM = "0000H" HIGH = "0001H".
Tipo Registro de Datos	(0000,0800)	US	1	Indica que hay un Registro de Datos presente en el mensaje. Debe ser cualquier valor distinto de 0101H (Nulo)
UID Instancia SOP	(0000,1000)	UI	1	UID de la Instancia SOP que se quiere almacenar
Título Entidad C-MOVE	(0000,1030)	AE	1	Contiene el Título de la Entidad de Aplicación que invocó el servicio C-MOVE que ha originado la suboperación C-STORE
ID del Mensaje C-MOVE	(0000,1031)	US	1	Contiene el ID del Mensaje CMOVE-RQ que ha originado la suboperación C-STORE
Registro de Datos	—	—	—	Conjunto de Datos específico de la Aplicación.

**Tabla 5.13.** Campos del Mensaje C-STORE-RSP

<b>Campo</b>	<b>Etiqueta</b>	<b>VR</b>	<b>VM</b>	<b>Descripción</b>
Longitud Grupo	(0000,0000)	UL	1	Número par de bytes desde el final del campo Valor hasta el comienzo del siguiente grupo.
UID Clase SOP	(0000,0002)	UI	1	UID de la Clase SOP a la que pertenece la Instancia SOP que se quiere almacenar.
Comando	(0000,0100)	US	1	Debe ser "0001H" para el mensaje C-STORE-RQ.
ID Mensaje Respondido	(0000,0120)	US	1	Debe ser el valor del campo "ID del Mensaje" del mensaje C-STORERQ.
Tipo Registro de Datos	(0000,0800)	US	1	Debe contener el valor "0101H" (Nulo), indicando que no hay un Registro de Datos presente.
Estado	(0000,0900)	US	1	El valor de este campo depende del tipo de estado.

Una vez recibido el mensaje C-STORE-RQ, la máquina de protocolo DIMSE-C deberá lanzar una primitiva de indicación al usuario DIMSE realizador. Lo mismo ocurrirá en el sentido contrario. Cuando la máquina de protocolo DIMSE-C reciba la primitiva de respuesta, lanzada por el usuario realizador, deberá:

- ♣ Construir un mensaje del tipo C-STORE-RSP.
- ♣ Mandar el mensaje utilizando el servicio P-DATA.

Una vez recibido el mensaje C-STORE-RSP, la máquina de protocolo DIMSE-C deberá lanzar una primitiva de confirmación al usuario DIMSE iniciador.

## 5.12. Servicios de Capa Superior

Los Servicios de Capa Superior son un subconjunto de los servicios ofrecidos por ACSE y la Capa de Presentación OSI, necesarios para soportar la comunicación entre Entidades de Aplicación OSI. Los Servicios de que se compone los tenemos en la tabla de tabla 5.14.

**Tabla 5.14.** Servicios de Capa Superior

Servicio	Tipo
A-ASSOCIATE	Confirmado
A-RELEASE	Confirmado
A-ABORT	No Confirmado
A-P-ABORT	Iniciado por Proveedor
P-DATA	No Confirmado

### 5.12.1. Servicio A-ASSOCIATE

El establecimiento de una asociación entre dos Entidades de Aplicación se realiza a través de las primitivas de petición, indicación, respuesta y confirmación del Servicio ACSE A-ASSOCIATE, de la manera que se puede observar en la figura 5.11.

Los parámetros requeridos por este servicio están recogidos en la tabla 5.15.



**Figura 5.11.** Servicio A-ASSOCIATE

**Tabla 5.15.** Parámetros del Servicio A-ASSOCIATE

Parámetro A-ASSOCIATE	Pet.	Ind.	Resp.	Conf.
application context name	M	M(=)	M	M(=)
calling AE title	M	M(=)	M	M(=)
called AE title	M	M(=)	M	M(=)
user information	M	M(=)	M	M(=)
result			M	M(=)
result source				M
diagnostic			U	C(=)
calling presentation address	M	M(=)		
calles presentation address	M	M(=)		
presentation context definition list	M	M(=)		
presentation context definition list result			M	M(=)

A continuación, vemos los parámetros con más detalle:

- ♣ *Application context name:* identifica el Contexto de Aplicación propuesto por el usuario iniciador. El usuario receptor debe devolver el mismo u otro nombre, siendo éste el que se utilice en la Asociación.
- ♣ *Calling AE title:* identifica a la Entidad de Aplicación que contiene al usuario iniciador del servicio A-ASSOCIATE.
- ♣ *Called AE title:* identifica a la Entidad de Aplicación que contiene al usuario receptor del servicio A-ASSOCIATE.
- ♣ *User information:* debe ser utilizado tanto por el iniciador como por el receptor para incluir información de usuario de las Entidades de Aplicación. El significado dependerá del Contexto de Aplicación que acompaña a la primitiva.

- ♣ *Result*: este parámetro debe ser proporcionado por el receptor de la petición A-ASSOCIATE, por el Proveedor de Servicios ACSE o por el Proveedor de Servicios de Capa Superior. Indica el resultado del uso del servicio A-ASSOCIATE. Toma uno de los siguientes valores:

- aceptado
- rechazado (permanente)
- rechazado (transitorio)

- ♣ *Result Source*: el valor de este parámetro lo proporciona el Proveedor de Servicios de Capa Superior. Identifica al origen del parámetro anterior. Toma uno de los siguientes valores:

- Proveedor de Servicios de Capa Superior
- Proveedor de Servicios ACSE
- Proveedor de Servicios de Presentación

- ♣ *Diagnostic*: sólo debe ser usado si el parámetro ‘Result’ tiene el valor ‘Rechazado (permanente)’ o ‘Rechazado (transitorio)’. Si el parámetro ‘Result Source’ tiene el valor ‘Proveedor de Servicios de Capa Superior’, este parámetro puede tomar uno de estos valores:

- causa no proporcionada
- nombre de contexto de aplicación no soportado
- título entidad de aplicación llamante no reconocido
- título entidad de aplicación llamada no reconocido

Si el parámetro ‘Result Source’ tiene el valor ‘Proveedor de Servicios ACSE’, este parámetro puede tomar uno de estos valores:

- causa no proporcionada
- versión capa superior no coincide

Si el parámetro ‘Result Source’ tiene el valor ‘Proveedor de Servicios de Presentación’ puede tomar uno de estos valores:

- causa no proporcionada

- congestión temporal
  - límite local excedido
  - dirección de presentación llamada desconocida
  - versión del protocolo de presentación no soportado
  - punto de acceso al servicio de presentación (SAP) no disponible
- 
- ♣ *Calling presentation address*: dirección de Presentación llamante. Debe contener una dirección no ambigua dentro del direccionamiento global de la red. Puede ser una dirección de presentación OSI o una dirección TCP/IP.
  - ♣ *Called presentation address*: dirección de Presentación llamada. Debe contener una dirección no ambigua dentro del direccionamiento global de la red. Puede ser una dirección de presentación OSI o una dirección TCP/IP.
  - ♣ *Presentation context definition list*: consiste en una lista con uno o más contextos de presentación. Cada elemento de esa lista contendrá tres componentes: la identificación del contexto de presentación, un nombre de una Sintaxis Abstracta, y una lista de uno o más nombres de Sintaxis de Transferencia.
  - ♣ *Presentation context definition result list*: indica la aceptación o rechazo de cada contexto de aplicación propuesto en el parámetro anterior. Consiste en una lista de resultados, que se corresponden uno a uno con los contextos de presentación propuestos. Los valores que puede tomar cada uno de lo resultado son:
    - aceptado
    - rechazado por el usuario
    - rechazado por el proveedor

Además de estos parámetros, hay otros que o bien no se utilizan en esta versión del Estándar DICOM o que llevan siempre un valor fijo, por lo que no se han incluido aquí.

### 5.12.2. Servicio A-RELEASE

La liberación ordenada de una asociación entre dos Entidades de Aplicación se realiza a través de las primitivas de petición, indicación, respuesta y confirmación del Servicio ACSE A-RELEASE.



Figura 5.12. Servicio A-RELEASE

Los parámetros requeridos por este servicio son los siguientes:

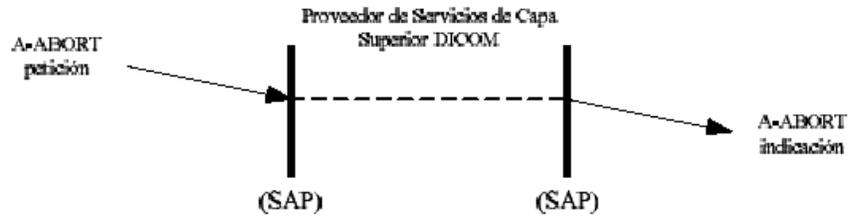
Tabla 5.16. Parámetros del Servicio A-RELEASE

Parámetro A-RELEASE	Pet.	Ind.	Resp.	Conf.
reason	UF	UF(=)	UF	UF(=)
user information	NU	NU(=)	NU	NU(=)
result			MF	MF(=)

- ♣ *Reason*: identifica el nivel de urgencia de la petición. Siempre debe llevar el valor "Normal".
- ♣ *Result*: siempre toma el valor "Afirmativo".

### 5.12.3. Servicio A-ABORT

El Servicio ACSE A-ABORT debe ser usado por alguna de las dos Entidades de Aplicación para liberar de forma anormal la asociación. Debe ser un servicio no confirmado, y se lleva a cabo a través de las primitivas de petición e indicación del Servicio ACSE A-ABORT.



**Figura 5.13.** Servicio A-ABORT

Los parámetros requeridos por este servicio son los siguientes:

**Tabla 5.17.** Parámetros del Servicio A-ABORT

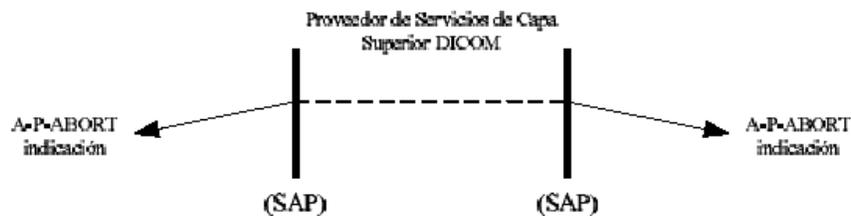
Parámetro A-ABORT	Petición	Indicación
abort source		M
user information	NU	NU(=)

♣ *Abort source*: indica la causa del inicio de la liberación abrupta. Debe tomar uno de los siguientes valores:

- Proveedor de Servicios de Capa Superior
- Proveedor de Servicios ACSE

#### 5.12.4. Servicio A-P-ABORT

El Servicio ACSE A-P-ABORT debe ser usado por el Proveedor de Servicios de Capa Superior para liberar de forma anormal la asociación debido a problemas en los servicios de la Capa de Presentación e inferiores. Es por tanto un servicio iniciado por el Proveedor de Servicios, y hace uso únicamente de la primitiva de indicación.



**Figura 5.14.** Servicio A-P-ABORT

Los parámetros requeridos por este servicio son los siguientes:

**Figura 5.18.** Parámetros del Servicio A-P-ABORT

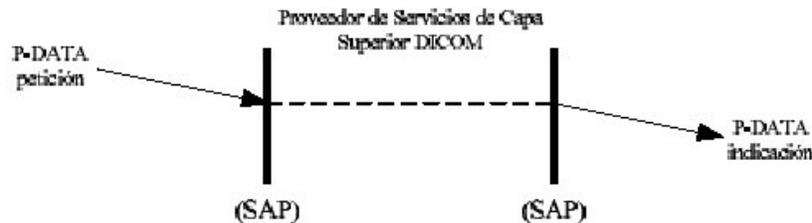
Parámetro A-P-ABORT	Indicación
provider reason	P

♣ *Provider reason*: indica las causas del inicio de la liberación abrupta. Debe tomar uno de los siguientes valores:

- causa no especificada
- PDU no reconocida
- PDU no esperada
- Primitiva del servicio de sesión no esperada
- parámetro de PDU no reconocido
- parámetro de PDU no esperado
- valor de parámetro de PDU incorrecto

### 5.12.5. Servicio P-DATA

El Servicio ACSE P-DATA debe ser usado por las dos Entidades de Aplicación para intercambiar información de aplicación, es decir, mensajes DICOM. Una asociación proporciona intercambio bidireccional de primitivas de petición e indicación.



**Figura 5.15.** Servicio P-DATA

Los parámetros requeridos por este servicio son los siguientes:

**Tabla 5.19.** Parámetros del Servicio P-DATA

Parámetro P-DATA	Petición	Indicación
presentation data value list	M	M(=)

### 5.13. Protocolo de Capa Superior para TCP/IP

Todo lo que hemos visto hasta se aplica de la misma forma en el caso de utilizar la torre de protocolos TCP/IP en lugar de OSI. Para ello, DICOM agrupa los Servicios de Capa Superior OSI en una capa denominada ‘Capa Superior DICOM’ (DUL, *DICOM Upper Layer*), que realiza las mismas funciones. Las diferencias más significativas aparecen a la hora de establecer las asociaciones entre entidades, ya que habrá que adaptarse a los mecanismos definidos en TCP/IP.

Hay una relación uno a uno entre una Conexión de Transporte TCP con una Asociación de Capa Superior, por lo tanto, son aplicables las siguientes reglas:

- ♣ Cada Asociación de Capa Superior debe ser soportada por una y sólo una Conexión de Transporte TCP.
- ♣ Cada Conexión de Transporte TCP soportará una y sólo una Asociación de Capa Superior.

Para establecer una conexión TCP, debe utilizarse algún puerto, que hará las veces de selector de transporte. Por tanto, una Entidad de Capa Superior DICOM se identificará en un determinado sistema en la red por un número de puerto único en el ámbito de ese sistema. El número de puerto de la Entidad remota a la que se conecta la Entidad local deberá ser, por tanto, configurable.

Cuando una Entidad DICOM desea establecer una Asociación, deberá lanzar una primitiva de petición de TRANSPORT CONNECT al Servicio de Transporte TCP. Una vez se recibe la confirmación, indicando que la Conexión de Transporte está establecida, se procederá al envío de la PDU A-ASSOCIATE-RQ.