

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

T.808

(01/2005)

SERIE T: TERMINALES PARA SERVICIOS DE
TELEMÁTICA

**Tecnología de la información – Sistema de
codificación de imágenes JPEG 2000:
Herramientas para la interactividad, interfaces
de programas de aplicación y protocolos**

Recomendación UIT-T T.808

Tecnología de la información – Sistema de codificación de imágenes JPEG 2000: Herramientas para la interactividad, interfaces de programas de aplicación y protocolos

Resumen

Esta Recomendación | Norma Internacional tiene como fin proporcionar un protocolo de red que permita la transmisión interactiva progresiva de datos y archivos codificados JPEG 2000 desde un servidor hasta un cliente. Gracias a este protocolo, un cliente puede pedir solamente las partes de una imagen (ya sea por región, calidad o nivel de resolución) que necesita. Asimismo, este protocolo facilita al cliente el acceso a metadatos u otro contenido del fichero.

Orígenes

La Recomendación UIT-T T.808 fue aprobada el 8 de enero de 2005 por la Comisión de Estudio 16 (2005-2008) del UIT-T por el procedimiento de la Recomendación UIT-T A.8. Se publica también un texto idéntico como Norma Internacional ISO/CEI 15444-9.

PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución 1 de la AMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

La observancia de esta Recomendación es voluntaria. Ahora bien, la Recomendación puede contener ciertas disposiciones obligatorias (para asegurar, por ejemplo, la aplicabilidad o la interoperabilidad), por lo que la observancia se consigue con el cumplimiento exacto y puntual de todas las disposiciones obligatorias. La obligatoriedad de un elemento preceptivo o requisito se expresa mediante las frases "tener que, haber de, hay que + infinitivo" o el verbo principal en tiempo futuro simple de mandato, en modo afirmativo o negativo. El hecho de que se utilice esta formulación no entraña que la observancia se imponga a ninguna de las partes.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 2005

Reservados todos los derechos. Ninguna parte de esta publicación puede reproducirse por ningún procedimiento sin previa autorización escrita por parte de la UIT.

ÍNDICE

	<i>Página</i>
1 Alcance.....	1
2 Referencias normativas	1
3 Definiciones	1
3.1 Definiciones de la Parte 1 de JPEG 2000.....	2
3.2 Definiciones relacionadas con el HTTP.....	2
3.3 Definiciones relativas al JPIP.....	2
3.4 Símbolos.....	3
4 Abreviaturas	5
5 Convenios.....	6
5.1 Reglas ABNF	6
5.2 Reglas ABNF de formato de fichero.....	6
5.3 Cómo entender las descripciones gráficas de las casillas (informativo)	6
6 Descripción general.....	7
6.1 Protocolo JPIP.....	7
6.2 Objetivo.....	8
7 Conformidad.....	9
Anexo A (normativo) – Los tipos de medios tren JPP y tren JPT	10
A.1 Introducción	10
A.2 Estructura de encabezamiento de mensaje	11
A.3 Bins de datos	13
A.4 Convenios para el análisis gramatical y la entrega de trenes JPP y JPT (informativo).....	21
A.5 Convenios para el interfuncionamiento de trenes JPP o JPT (informativo).....	21
Anexo B (normativo) – Sesiones, canales, modelo caché y conjuntos modelo	23
B.1 Compartición entre las peticiones dentro de una sesión y las peticiones sin estado.....	23
B.2 Canales y sesiones.....	23
B.3 Gestión de modelo caché.....	24
B.4 Interrogación y manipulación de conjuntos modelo.....	24
Anexo C (normativo) – Petición de cliente.....	25
C.1 Sintaxis de petición	25
C.2 Campos de identificación de objetivo	26
C.3 Campos para el trabajo con sesiones y canales	28
C.4 Campos de petición de ventana seleccionada.....	29
C.5 Campos de petición de metadatos	37
C.6 Campos de petición de limitación de datos	40
C.7 Campos de petición de control de servidor	41
C.8 Campos de petición de gestión de caché	43
C.9 Parámetros de petición de carga.....	49
C.10 Capacidad de cliente y campos de petición de preferencia	49
Anexo D (normativo) – Señalización de respuesta del servidor	56
D.1 Sintaxis de la respuesta	56
D.2 Encabezamiento de respuesta JPIP.....	57
D.3 Datos de respuesta.....	62
Anexo E (normativo) – Carga de imágenes en el servidor	63
E.1 Introducción	63
E.2 Petición de carga	63
E.3 Respuesta del servidor.....	64
E.4 Fusión de datos en el servidor	64

Anexo F – Utilización de JPIP sobre HTTP	66
F.1 Introducción	66
F.2 Peticiones	66
F.3 Establecimiento de una sesión.....	67
F.4 Respuestas.....	68
F.5 Características del HTTP adicionales.....	69
F.6 HTTP y el campo de petición de longitud (informativo)	69
Anexo G (normativo) – Utilización de JPIP con peticiones HTTP y respuestas TCP.....	70
G.1 Introducción	70
G.2 Peticiones de cliente.....	70
G.3 Establecimiento de sesión	70
G.4 Respuestas del servidor	71
G.5 TCP y campo de petición de longitud (informativo).....	71
Anexo H (informativo) – Utilización de JPIP con transportes diversos	72
H.1 Introducción	72
H.2 Peticiones fiables con datos no fiables	72
H.3 Peticiones no fiables con datos no fiables.....	73
H.4 Sintaxis de petición y respuesta	74
H.5 Establecimiento de sesiones.....	74
Anexo I (informativo) – Indexación de ficheros JPEG 2000 para JPIP.....	75
I.1 Introducción (informativo).....	75
I.2 Identificación del uso de casillas índice JPIP en la lista de compatibilidad del formato de fichero JPEG 2000	76
I.3 Casillas definidas	76
I.4 Relación de los índices de trenes de código con los trenes de código.....	85
I.5 Restricciones relativas a la ubicación (informativo)	85
Anexo J (informativo) – Registro de ampliaciones a esta Recomendación Norma Internacional	86
J.1 Introducción al registro	86
J.2 Elementos de registro	86
J.3 Criterios de evaluación del registro.....	86
J.4 Elementos que pueden ampliarse por registro.....	86
J.5 Proceso de registro	87
J.6 Plazos para el proceso de registro	87
Anexo K (informativo) – Ejemplos de aplicación	88
K.1 Introducción	88
K.2 Utilización del JPIP con trenes de código en otros formatos de fichero	88
K.3 Técnicas de implementación de la parte losa	88
K.4 Técnicas de implementación basadas en recintos	89
K.5 Transcripción de protocolo JPIP	90
K.6 Utilización de JPIP con HTML.....	94
Anexo L (informativo) – Colección ABNF del JPIP	95
L.1 ABNF de petición JPIP	95
L.2 BNF de respuesta JPIP.....	102
Anexo M (informativo) – Declaración de patentes.....	105
Anexo N (informativo) – Bibliografía	106

FIGURAS

	<i>Página</i>
Figura 1 – Ejemplo de las figuras de descripción de casillas.....	7
Figura 2 – Ejemplo de figura de descripción de supercasillas	7
Figura 3 – Visión general del protocolo JPIP	8
Figura 4 – Pila de protocolo JPIP	8
Figura A.1 – Ejemplos de fichero JPEG 2000, bins de datos JPIP y relaciones de tren JPIP (tomado de G.J. Colyer and R.A. Clark, IEEE Trans. Consumer Electronics, 49 (2003), pp 850–854).....	10
Figura A.2 – Estructura de VBAS	11
Figura A.3 – Estructura del VBAS Bin-ID.....	11
Figura A.4 – Ejemplo de bin de datos de recinto.....	14
Figura A.5 – Ejemplo de notación de colores para bins de metadatos.....	15
Figura A.6 – Muestra de fichero JP2	16
Figura A.7 – Muestra de fichero JP2 dividido en tres bins de metadatos	16
Figura A.8 – Supercasilla con bin de metadatos referenciados	17
Figura A.9 – División no válida del fichero en bins de metadatos	18
Figura A.10 – Ejemplo de utilización de equivalentes de tren	19
Figura A.11 – Estructura de casilla marcador de posición.....	19
Figura C.1 – Región deseada dentro de una imagen.....	30
Figura C.2 – Región deseada con respecto a la rejilla de referencia submuestreada	30
Figura C.3 – Procedimiento de selección de casilla de especificación de espacio de color.....	52
Figura G.1 – Estructura de datos de la respuesta por una conexión http-tcp	71
Figura I.1 – Parte de un fichero JPEG 2000 de ejemplo que contiene casillas índice JPIP	76
Figura I.2 – Organización del contenido de una casilla de índice de tren de código	77
Figura I.3 – Organización del contenido de una casilla buscador de tren de código	78
Figura I.4 – Organización del contenido de una casilla declaración.....	78
Figura I.5 – Organización del contenido de una casilla índice matriz de fragmentos.....	79
Figura I.6 – Organización del contenido de una casilla de tabla de índice de encabezamiento.....	80
Figura I.7 – Organización del contenido de la casilla tabla de índice de parte losa.....	81
Figura I.8 – Organización del contenido de una casilla tabla de índice de encabezamiento de losa	81
Figura I.9 – Organización del contenido de una casilla tabla de índice de paquetes recinto	82
Figura I.10 – Organización del contenido de la casilla tabla de índice de encabezamiento de paquete	82
Figura I.11 – Organización del contenido de una casilla índice de fichero	83
Figura I.12 – Organización del contenido de una casilla buscador de fichero.....	83
Figura I.13 – Organización del contenido de una casilla intermediario.....	84
Figura I.14 – Organización del contenido de la casilla buscador de fichero.....	84

CUADROS

	<i>Página</i>
Cuadro A.1 – Indicación de VBAS adicional de Bin-ID	12
Cuadro A.2 – Identificadores de clase para las diversas clases de mensajes de bin de datos	12
Cuadro A.3 – Valores admitidos para el campo Banderas de una casilla marcador de posición	20
Cuadro C.1 – Opciones de aproximación	32
Cuadro C.2 – Banderas calificador de petición de metadatos	40
Cuadro C.3 – Fronteras de alineamiento basadas en el tipo de bin	41
Cuadro C.4 – Tipos válidos de retorno de imagen	42
Cuadro C.5 – Resumen de opción de descriptor de caché	46
Cuadro C.6 – Capacidades válidas del elemento <code>processing-capabilities</code>	50
Cuadro C.7 – Valores válidos de los parámetros <code>config-capability</code>	50
Cuadro C.8 – Preferencias para el manejo de ventanas seleccionadas	51
Cuadro C.9 – Preferencias de cliente de método de espacio de color	52
Cuadro C.10 – Preferencia de marcador de posición	54
Cuadro C.11 – Preferencias de secuenciamiento de tren de código	54
Cuadro D.1 – Valores válidos de <code>transport-param</code>	58
Cuadro D.2 – Códigos de razón definidos	62
Cuadro I.1 – Casillas definidas (informativo)	77
Cuadro I.2 – Valores del tipo de contenedor	78
Cuadro I.3 – Valores de la versión	80
Cuadro K.1 – Ejemplo de utilización de los campos auxiliares en un caso sencillo	89
Cuadro K.2 – Ejemplo de utilización de campos auxiliares en un caso más complejo	89

Introducción

La Rec. UIT-T T.800 | ISO/CEI 15444-1 (JPEG 2000) es una especificación que describe un sistema de compresión de imágenes que permite una gran flexibilidad, no solamente a dichos efectos sino también en lo que se refiere al acceso al tren de código, con el que hay una variedad de mecanismos para ubicar y extraer porciones de los datos de imagen comprimida, a fin de retransmitirlos, almacenarlos, presentarlos en pantalla o editarlos. Gracias a este acceso, es posible almacenar y recuperar los datos de una imagen comprimida adecuados para una determinada aplicación, sin que sea necesaria la decodificación.

En esta Recomendación | Norma Internacional se pretende proporcionar un protocolo de red a través del cual se puedan transmitir interactiva y progresivamente los datos y ficheros codificados JPEG 2000 de un servidor a un cliente. El cliente puede, utilizando este protocolo, pedir solamente las porciones de una imagen (bien sea por región, calidad o nivel de resolución) que se ajusten a sus necesidades. De igual manera, con este protocolo el cliente puede acceder los metadatos u otro contenido del fichero.

Toda organización que pretenda utilizar esta Recomendación | Norma Internacional ha de prestar mucha atención a su aplicabilidad.

La Unión Internacional de Telecomunicaciones (UIT), la Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (CEI) llaman la atención sobre el hecho de que la conformidad con esta Recomendación | Norma Internacional puede implicar la utilización de material ya patentado.

La UIT, la ISO y la CEI no se pronuncian sobre las pruebas, validez y alcance de estos derechos de autor.

El poseedor de dicha patente ha garantizado a la UIT, la ISO y la CEI que está dispuesto a negociar con candidatos de todo el mundo licencias en términos y condiciones razonables y no discriminatorios. A este fin, se ha registrado en la UIT, la ISO y la CEI la declaración del poseedor de este derecho de autor. En el anexo M se incluye la lista de empresas que pueden proporcionar más información al respecto.

De igual manera, cabe señalar la posibilidad de que algunos de los elementos de esta Recomendación | Norma Internacional tengan que ver con derechos de autor diferentes de aquellos identificados en el anexo M. No es responsabilidad de la UIT, la ISO y la CEI identificar ninguno de esos derechos.

**NORMA INTERNACIONAL
RECOMENDACIÓN UIT-T**

**Tecnología de la información – Sistema de codificación de imágenes JPEG 2000:
Herramientas para la interactividad, interfaces de programas
de aplicación y protocolos**

1 Alcance

En esta Recomendación | Norma Internacional se definen, de un modo extensible, las sintaxis y los métodos necesarios para interrogar y, facultativamente, modificar a distancia los trenes de código y ficheros JPEG 2000, conforme con las definiciones suministradas en las siguientes partes de ISO/CEI 15444:

- Rec. UIT-T T.800 | ISO/CEI 15444-1 y su definición de un tren de código JPEG 2000 y un formato de fichero JP2.
- La familia de formatos de fichero JPEG 2000 se define en otras partes de ISO/CEI 15444.

En esta Recomendación | Norma Internacional, la sintaxis y métodos definidos se denominan protocolo interactivo JPEG 2000, "JPIP", y las aplicaciones interactivas que lo utilizan se conocen como "sistemas JPIP".

En JPIP se especifica un protocolo compuesto por una serie estructurada de interacciones entre un cliente y un servidor, mediante la cual es posible intercambiar de manera eficiente metadatos de ficheros de imagen y estructuras de trenes de código de imagen parciales o totales. Esta Recomendación | Norma Internacional incluye definiciones de las semánticas y valores que han de intercambiarse, y se sugiere cómo hacerlos pasar a través de varias de las redes de transporte actuales.

Gracias al JPIP es posible efectuar de forma compatible y de varias maneras las siguientes tareas:

- el intercambio de capacidades;
- la negociación de capacidades que se han de utilizar en una sesión;
- la petición y transferencia de los siguientes elementos desde diversos contenedores, como por ejemplo los ficheros de la familia JPEG 2000, los trenes de código JPEG 2000 y otros ficheros de contenedor:
 - segmentos de datos selectivos,
 - estructuras selectivas y definidas,
 - partes de una imagen o sus metadatos relacionados.

2 Referencias normativas

Las siguientes Recomendaciones y Normas Internacionales contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación | Norma Internacional. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y Normas son objeto de revisiones, por lo que se preconiza que los participantes en acuerdos basados en la presente Recomendación | Norma Internacional investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y las Normas citadas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales actualmente vigentes. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T actualmente vigentes.

- Recomendación UIT-T T.800 (2002) | ISO/CEI 15444-1:2004, *Tecnología de la información – Sistema de codificación de imágenes JPEG 2000: Sistema de codificación básico.*
- Recomendación UIT-T T.801 (2002) | ISO/CEI 15444-2:2004, *Tecnología de la información – Sistema de codificación de imágenes JPEG 2000: Extensiones.*
- Recomendación UIT-T T.802 (2005) | ISO/CEI 15444-3:2005, *Tecnología de la información – Sistema de codificación de imágenes JPEG 2000: JPEG 2000 en movimiento*
- ISO/CEI 15444-6:2003, *Information technology – JPEG 2000 image coding system – Part 6: Compound image file format.*

ISO/CEI 15444-9:2005 (S)

- IETF RFC 768 (1980), *User Datagram Protocol*. Available from World Wide Web: <<http://www.ietf.org/rfc/rfc0768.txt>>.
- IETF RFC 793 (1981), *Transmission Control Protocol*. Available from World Wide Web: <<http://www.ietf.org/rfc/rfc0793.txt>>.
- IETF RFC 2046 (1996), *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. Available from World Wide Web: <<http://www.ietf.org/rfc/rfc2046.txt>>.
- IETF RFC 2234 (1997), *Augmented BNF for Syntax Specifications: ABNF*. Available from World Wide Web: <<http://www.ietf.org/rfc/rfc2234.txt>>.
- IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*. Available from World Wide Web: <<http://www.ietf.org/rfc/rfc2396.txt>>.
- IETF RFC 2616 (1999), *Hypertext Transfer Protocol – HTTP/1.1*. Available from World Wide Web: <<http://www.ietf.org/rfc/rfc2616.txt>>.

3 Definiciones

A los efectos de la presente Recomendación | Norma Internacional, se aplican las siguientes definiciones.

3.1 Definiciones de la Parte 1 de JPEG 2000

Las definiciones que figuran en la cláusula 3 de la Rec. UIT-T T.800 | ISO/CEI 15444-1:2004, y en la cláusula 3 de la Rec. UIT-T T.801 | ISO/CEI 15444-2:2004, también se aplican a la presente Recomendación | Norma Internacional.

3.2 Definiciones relacionadas con el HTTP

Las siguientes definiciones tienen como fin una conformidad con el HTTP/1.1. En caso de duda han de utilizarse éstas.

3.2.1 conexión: Circuito virtual de capa de transporte que se establece entre dos programas a efectos de comunicación.

3.2.2 entidad: Información que se transfiere en la cabida útil de una petición o respuesta, y que consta de metainformación en los campos encabezamiento de entidad y contenido en el cuerpo de entidad.

3.2.3 intermediario: Programa que funciona como servidor o como cliente a efectos de realizar peticiones a nombre de otros clientes. Las peticiones se atienden internamente o después de hacerlas pasar, con posible traducción, a otros servidores.

3.3 Definiciones relativas al JPIP

Las definiciones que se presentan a continuación son utilizadas en esta Recomendación | Norma Internacional. En algunos casos son diferentes de las que se utilizan en otras normas y/o Recomendaciones.

3.3.1 caché (lado de cliente): Memoria de almacenamiento de los bins de datos JPIP. Es posible que la caché del cliente sea limitada y, por ende, se deban purgar de vez en cuando los bins de datos JPIP allí almacenados.

3.3.2 almacenable en caché ("Cacheable"): Se dice que una respuesta es cacheable si es posible almacenar en la caché una copia del mensaje de respuesta a fin de utilizarla en la respuesta de peticiones subsiguientes. Aun si un recurso es cacheable, pueden existir otras restricciones relativas a la utilización de una copia almacenada para determinada petición.

3.3.3 modelo caché (lado del servidor): Porciones de los bins de datos disponibles, según el servidor, en la caché de cliente. El servidor puede añadir ítems a su estimación de la caché de cliente debido a que supone que se ha entregado con éxito o que se han recibido acusos de transmisión de datos, o como consecuencia de declaraciones de actualización del modelo caché.

3.3.4 canal: Mecanismo de agrupación de peticiones y respuestas mediante el cual se activa solamente una petición/respuesta a la vez dentro del grupo. Son necesarios múltiples canales para procesar múltiples peticiones y respuestas simultáneas.

3.3.5 cliente: Programa que establece conexiones con el fin de enviar peticiones.

3.3.6 región de imagen de tren de código: Intersección entre la imagen y la región definida por la traslación y el tamaño de región. Puede estar vacía (sin área).

3.3.7 bin de datos: Conjunto de bytes del mismo tipo de datos que se puede enviar parcialmente.

- 3.3.8 tren de código incremental:** Representación del tren de código como conjunto de bins de datos (de encabezamiento principal, de encabezamiento de losa, de recinto o de losa) cuyo identificador de tren de código sea idéntico.
- 3.3.9 tabla de índices JPIP:** Tabla de formato de fichero que proporciona información acerca de la ubicación de porciones de un fichero o tren de código.
- 3.3.10 objetivo lógico:** Representación específica de un determinado recurso designado original, o una gama de bytes de él, a la cual se dirige la petición JPIP. Se puede transcodificar a partir del recurso designado original.
- 3.3.11 mensaje:** Conjunto de bytes de un solo bin de datos cuyo encabezamiento los identifica tanto a ellos como al bin de datos.
- 3.3.12 tren de código en bruto:** Dícese de una representación del tren de código como un solo bin de metadatos.
- 3.3.13 petición:** Grupo de campos y valores que se envían desde el cliente hasta el servidor para obtener porciones de una imagen o metadatos.
- 3.3.14 recurso:** Un objeto o servicio de datos de red que se puede identificar mediante un URI. Un objetivo HTTP.
- 3.3.15 respuesta:** Bytes que se envían desde el servidor hacia el cliente tras haber recibido una petición.
- 3.3.16 servidor:** Programa de aplicación que acepta conexiones con el fin de atender peticiones mediante el envío de respuestas. Todo programa debe poder ser a la vez cliente y servidor; estos términos se utilizan solamente para referirse a la función ejecutada por el programa en lo relativo a determinada conexión, y no a las capacidades del programa en general.
- 3.3.17 sesión:** Conjunto de peticiones y respuestas que se aplican al mismo recurso, y para las que el servidor mantiene un modelo caché.
- 3.3.18 basado en sesión:** Siempre que el servidor mantenga un modelo caché.
- 3.3.19 sin estado:** Una petición para la cual el servidor no utiliza un modelo caché para establecer la respuesta.
- 3.3.20 objetivo:** Identificación lógica de datos JPIP. Nombre del objetivo principal (suele ser el nombre de un fichero en el servidor).
- NOTA – Los ficheros o trenes de código JPEG 2000 pueden estar disponibles en muchas representaciones (por ejemplo, tipo de retorno o tamaño de recinto) o variar de otras maneras, cada una identificada como un solo objetivo lógico.
- 3.3.21 encabezamiento de losa:** Todos los encabezamientos de parte losa de una losa específica.
- 3.3.22 ventana seleccionada:** Porción de los datos de imagen escogida por el cliente, y que se expresa mediante la combinación de los siguientes campos que aparecen en la petición: tamaño de región, traslación, tamaño de cuadro, tren de códigos, contexto de tren de código, velocidad de muestreo, ROI y Capas. Suele ser menor que el total de los datos de imagen. Cuando es implícita más no especificada, se habrá de tomar de todos los datos de imagen del objetivo lógico.

3.4 Símbolos

A los efectos de esta Recomendación | Norma Internacional, se aplican los siguientes símbolos. También se aplican los símbolos definidos en la cláusula 4 de la Rec. UIT-T T.800 | ISO/CEI 15444-1:2004, y en la cláusula 4 de la Rec. UIT-T T.801 | ISO/CEI 15444-2:2004.

c	Índice (empezando desde el número 0), de la componente de imagen a la que pertenece el recinto
fx	Tamaño de cuadro en el eje x para una ventana seleccionada por petición de cliente
fy	Tamaño de cuadro en el eje y para una ventana seleccionada por petición de cliente
fx'	Tamaño de cuadro en el eje x, necesario para disponer de una resolución adecuada del tren de códigos
fy'	Tamaño de cuadro en el eje y, necesario para disponer de una resolución adecuada del tren de códigos
fx''	Tamaño de trama en el eje x modificado por jpx, necesario para tener una resolución adecuada
fy''	Tamaño de trama en el eje y modificado por jpy, necesario para tener una resolución adecuada

ISO/CEI 15444-9:2005 (S)

H_{cod}	Altura del tren de código como aparece en la tabla de encabezamiento de imagen (ihdr) (véase anexo I.5.3.1 de la Rec. UIT-T T.800 ISO/CEI 15444-1:2004)
H_{comp}	Altura del resultado compuesto, suministrada en la casilla de opciones de composición JPX (véase anexo M.11.10.1 de la Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
H_{reg}	Altura de la capa de composición, tal como figura en la rejilla de registro de dicha capa
HS_{inst}	Altura recortada
Ht_{inst}	Altura compuesta
l	Identificador único del recinto en su tren de código
N_L	Cantidad de niveles de descomposición
num_components	Cantidad de componentes codificadas
num_tiles	Cantidad de losas en el tren de código
ox	Traslación en el eje x de la ventana seleccionada por petición de cliente
ox'	Traslación en el eje x, necesaria para la región adecuada de tren código
ox''	Traslación en el eje x modificada por jpx, necesaria para tener la región adecuada
oy	Traslación en el eje y, necesaria para tener la ventana seleccionada por petición de cliente
oy'	Traslación en el eje y, necesaria para tener la región adecuada de tren de código
oy''	Traslación del eje y modificada por jpx, necesaria para tener la región adecuada
r	Nivel de resolución
s	Número de secuencia que identifica el recinto dentro de su componente de losa
sx	Tamaño en el eje x de la ventana seleccionada por petición de cliente
sx'	Tamaño en el eje x, necesario para tener la región adecuada de tren de código
sx''	Tamaño en el eje x modificado por jpx, necesario para tener la región adecuada
sy	Tamaño en el eje y de la ventana seleccionada por petición de cliente
sy'	Tamaño en el eje y, necesario para tener la región adecuada de tren de código
sy''	Tamaño en el eje y modificado por jpx, necesario para tener la región adecuada
t	Índice (que empieza desde 0) de la losa a la que pertenece el recinto
W_{cod}	Ancho de tren de código como figura en la caja de encabezamiento de imagen (ihdr) (véase anexo I.5.3.1 de la Rec. UIT-T T.800 ISO/CEI 15444-1:2004)
W_{comp}	Ancho del resultado compuesto, suministrado en la caja de opciones de composición JPX (véase anexo M.11.10.1 de la Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
W_{reg}	Ancho de la capa de composición, como figura en la rejilla de registro de capa de composición
WS_{inst}	Ancho recortado
Wt_{inst}	Ancho compuesto
XC_{inst}	Traslación de corte en el eje x, suministrada a través de la instrucción pertinente (véase anexo M.11.10.2.1 de la Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
XO_{inst}	Traslación compuesta en el eje x, descrita mediante la instrucción pertinente de composición (véase anexo M.11.10.2.1 de la Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
XO_{reg}	Traslación de registro de tren de código en el eje x
XO_{siz}	Traslación horizontal a partir del origen de la rejilla de referencia del segmento marcador SIZ del tren de código pertinente

XR_{reg}	Factor de muestreo de registro de tren de código en el eje x, descrito a partir del inicio de cualquier casilla de registro de tren de código (véase anexo M.11.7.7 de la Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
Xsiz	Ancho de la rejilla de referencia del segmento marcador SIZ del tren de código pertinente
XS_{reg}	Precisión de registro en el eje x, descrita al inicio de cualquier casilla de registro de tren de código (véase anexo M.11.7.7 de la Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
YC_{inst}	Traslación de recorte en el eje y, suministrada a través de las instrucciones del caso (véase anexo M.11.10.2.1 de la Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
YO_{inst}	Traslación de composición en el eje y, descrita a través de la instrucción de composición pertinente (véase anexo M.11.10.2.1 de la Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
YO_{reg}	Traslación de registro de tren de código en el eje y
YOsiz	Traslación vertical a partir del origen de la rejilla de referencia del segmento marcador SIZ del tren de código pertinente
YR_{reg}	Factor de muestreo de registro de tren de código en el eje y, descrito al inicio de cualquier casilla de registro de tren de código (véase anexo M.11.7.7 de la Rec. UIT-T T.801 ISO/CEI 15444-2:2004)
Ysiz	Altura de la rejilla de referencia del segmento marcador SIZ del tren de código pertinente
YS_{reg}	Precisión de registro en el eje y, descrita al inicio de cualquier casilla de registro de tren de código (véase anexo M.11.7.7 de la Rec. UIT-T T.801 ISO/CEI 15444-2:2004)

4 Abreviaturas

A los efectos de esta Recomendación | Norma Internacional se utilizan las siguientes abreviaturas.

ABNF	Forma aumentada de Backus-Naur (<i>augmented Backus-Naur form</i>)
DICOM	Imágenes y comunicaciones digitales en medicina (<i>digital imaging and communications in medicine</i>)
DWT	Transformada en ondícula discreta (<i>discrete wavelet transformation</i>)
EOR	Fin de respuesta (<i>end of response</i>)
HTML	Lenguaje de marcaje hipertexto (<i>hypertext markup language</i>)
IP	Protocolo Internet (<i>Internet protocol</i>)
JP3D	Parte 10 de JPEG 2000: 3-D y datos de coma flotante (<i>JPEG 2000 part 10: 3-d and floating point data</i>)
JPIP	Protocolo interactivo JPEG 2000 (<i>JPEG 2000 interactive protocol</i>)
JPP	Recinto JPIP (<i>jpip precinct</i>)
JPSEC	Parte 8 de JPEG 2000: JPEG-2000 seguro (<i>JPEG 2000 part 8: secure JPEG 2000</i>)
JPT	Parte losa JPIP (<i>JPIP tile-part</i>)
JPWL	Parte 11 de JPEG 2000: inalámbricos (<i>JPEG 2000 part 11: wireless</i>)
JTC 1	Comité Técnico Mixto 1 (<i>Joint Technical Committee 1</i>)
MTF	Función de transferencia de modulación (<i>modulation transfer function</i>)
PDF	Formato de documento portable (<i>portable document format</i>)
SC 29	Subcomité 29 (<i>Sub-Committee 29</i>)
SVG	Gráficos de vector escalable (<i>scalable vector graphics</i>)
TCP	Protocolo de control de transmisión (<i>transmission control protocol</i>)
UDP	Protocolo de datagrama de usuario (<i>user datagram protocol</i>)
UUID	Identificador único universal (<i>universal unique identifier</i>)
VBAS	Segmento alineado de byte de longitud variable (<i>variable-length byte aligned segment</i>)
WG 1	Grupo de Trabajo 1 (<i>Working Group 1</i>)

XHTML	Lenguaje de marcaje de hipertexto extensible (<i>extensible hypertext markup language</i>)
XML	Lenguaje de marcaje extensible (<i>extensible markup language</i>)

5 Convenios

5.1 Reglas ABNF

En esta Recomendación | Norma Internacional se utiliza la notación ABNF que se define en el RFC 2234, incluyendo el núcleo de reglas de sintaxis ABNF, a saber, ALPHA (letras), CR (retorno de carro), CRLF (nueva línea según la norma Internet), CTL (caracteres de control), DIGIT (números decimales), HEXDIG (números hexadecimales), LF (salto de línea), LWSP (espacio en blanco lineal) y SP (espacio). A efectos de esta Recomendación | Norma Internacional, también se aplican las siguientes reglas ABNF.

```
NZDIGIT = %x31-39           ; 1-9
UPPER = %x41-5A            ; A-Z
LOWER = %x61-7A           ; a-z
UINT = 1*DIGIT
NONZERO = "*"0" NZDIGIT *DIGIT
UINT-RANGE = UINT ["-" [UINT]]
UFLOAT = 1*DIGIT [ "." 1*DIGIT ]
ENCODED-CHAR = "%" HEXDIG HEXDIG
UUID = 16(HEXDIG)
TOKEN = 1*(ALPHA / DIGIT / "." / "_" )
```

En esta Recomendación | Norma Internacional también se define PATH, que representa un fichero o una ruta de acceso. En general, los valores PATH pueden incluir cualquier carácter, aunque es posible que para determinada arquitectura de servidor, éste rechace todos los caracteres que no sean válidos en su caso particular. Además, se debe codificar PATH adecuadamente conforme a la tecnología de transporte.

UINT-RANGE especifica un intervalo de valores enteros, cuyo primer entero indica el inicio del intervalo. Si se especifican dos valores, éstos indican respectivamente el inicio y fin del intervalo. Si solamente se especifica el primero y el carácter "-", el intervalo contiene todos los valores mayores o iguales que dicho valor.

Cuando un elemento ABNF vaya precedido de un valor numérico, quiere decir que se repite el parámetro que viene después tantas veces como lo indique el valor, sin que haya espacios entre cada una de ellos.

El concepto "1#" indica una o varias repeticiones separadas por comas del parámetro que viene a continuación.

El concepto "1\$" indica una o varias repeticiones, separadas por punto y coma, del parámetro que sigue.

5.2 Reglas ABNF de formato de fichero

```
compatibility-code = 4(ALPHA / DIGIT / "_" / ENCODED-CHAR)
box-type = 4(ALPHA / DIGIT / "_" / ENCODED-CHAR)
box-type-list = "*" / 1$(box-type)
```

box-type especifica los cuatro caracteres del tipo de casilla. Para cada tipo de casilla, si el carácter es alfanumérico (A..Z, a..z o 0..9), se escribe directamente en la cadena. Si es un espacio (0x20), se ha de codificar mediante el carácter subrayado ("_"). En los demás casos, se escribe una cadena de 3 caracteres en su lugar, conformada por el carácter ("%") seguido de dos números hexadecimales que representan el valor del carácter tomado del tipo de casilla, en hexadecimal. Se codifica el compatibility-code de la misma manera que el box-type.

box-type-list especifica una lista de tipos de casilla. Si el valor del campo box-type-list es "*", el campo es un comodín que se refiere a todos los tipos de casilla.

5.3 Cómo entender las descripciones gráficas de las casillas (informativo)

La descripción de cada casilla viene acompañada de una figura en la que se muestra el orden y las relaciones de los parámetros en ella. En la figura 1 se presenta un ejemplo de este tipo de figura: se utiliza un rectángulo para indicar el

parámetro en la casilla, cuyo ancho es proporcional a la cantidad de bytes en el parámetro. Un parámetro de tamaño variable se señala mediante un rectángulo sombreado (con rayas diagonales). Dos parámetros acompañados de superíndices y con una zona gris entre ellos indican que hay un grupo de parámetros entre los dos. Una secuencia de dos grupos de parámetros múltiples acompañados de superíndices y separados por una zona gris indica la ejecución de dicho grupo de parámetros (un conjunto de cada parámetro seguido por el próximo conjunto de cada parámetro en el grupo). Se utilizarán rectángulos punteados para señalar los parámetros o casillas facultativos.

Después de la figura se presenta una lista en la que se describe el significado de cada parámetro en la casilla. Cuando haya parámetros repetidos, se define la longitud y naturaleza del grupo de parámetros. Por ejemplo, en la figura 1 los parámetros A, B, C y D tienen longitudes de 8, 16, 32 bits y variable, respectivamente. La notación E^0 y E^{N-1} implica que existen N parámetros diferentes, E^i , en una fila. El grupo de parámetros F^0 y F^{M-1} , y G^0 y G^{M-1} especifica que habrá en la casilla F^0 , seguido por G^0 , seguido por F^1 y G^1 , y así sucesivamente hasta F^{M-1} y G^{M-1} (en total M ejemplares de cada parámetro). Asimismo, como el campo D es facultativo es posible que no se encuentre en esta casilla.

Además, se utilizarán los puntos suspensivos (...) en una figura que describa los contenidos de una supercasilla para indicar que no se han definido específicamente los contenidos del fichero entre dos casillas. A menos que se especifique lo contrario, toda casilla (o secuencia de casillas) especificada por la definición de dicha casilla puede ir en el lugar señalado con puntos suspensivos.

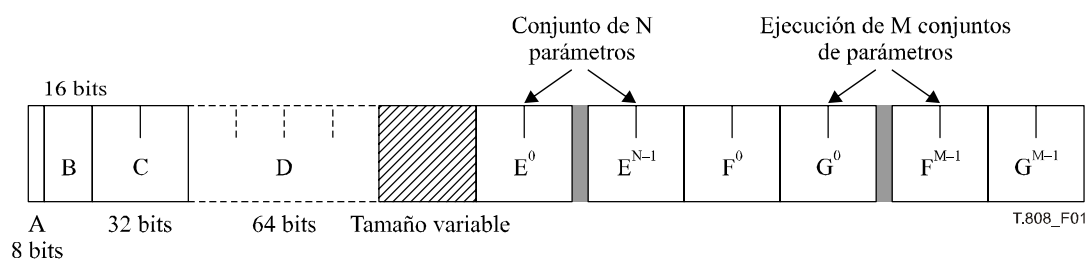


Figura 1 – Ejemplo de las figuras de descripción de casillas

Por ejemplo, la supercasilla mostrada en la figura 2 debe incluir una casilla AA y una BB tras ella. No obstante, puede haber muchas otras casillas entre las casillas AA y BB. En el anexo I.8 de la Rec. UIT-T T.800 | ISO/CEI 15444-1:2004 se indica cómo tratar el caso de las casillas desconocidas.

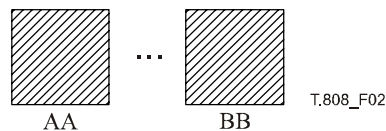


Figura 2 – Ejemplo de figura de descripción de supercasillas

6 Descripción general

6.1 Protocolo JPIP

En esta Recomendación | Norma Internacional se describen las sintaxis y métodos utilizados cuando un cliente accede a imágenes y datos relacionados con imágenes, comprimidos mediante JPEG 2000, que se encuentran en un servidor que acepta el JPIP. También se permite obtener, a través de múltiples transportes cliente/servidor, la flexibilidad y funcionalidad pretendidas en la Rec. UIT-T T.800 | ISO/CEI 15444-1:2004.

En JPIP se define el protocolo interactivo necesario para lograr un intercambio eficaz de imágenes JPEG 2000 y datos relativos a ellas. En este protocolo se definen las interacciones cliente-servidor basándose en una petición de cliente y una respuesta de servidor, como se muestra en la figura 3. En esta Recomendación | Norma Internacional se definen las peticiones de cliente JPIP y las respuestas de servidor JPIP. Como posibles ejemplos de transportes para JPIP se muestran los HTTP/1.1 (RFC 2616), TCP (RFC 793) y UDP (RFC 768). Para definir la resolución, tamaño, ubicación, componentes, capas y otros parámetros de la imagen y datos relativos a ella que solicita el cliente, éste utiliza una petición de ventana seleccionada, tras lo cual el servidor responde entregando una imagen y datos relativos a ella mediante trenes basados en recintos, trenes basados en losas o imágenes completas. Del mismo modo, es posible que el cliente y el servidor negocien sus capacidades y limitaciones. El cliente puede pedir al servidor información acerca de una imagen conforme a lo que se define en las tablas de índices JPIP, con lo cual puede refinar su petición de ventana

seleccionada adecuándola a parámetros específicos de la imagen (por ejemplo, petición de intervalo de bytes). El modelo caché del servidor se basa en las capacidades definidas por el cliente y en el estado de la sesión.

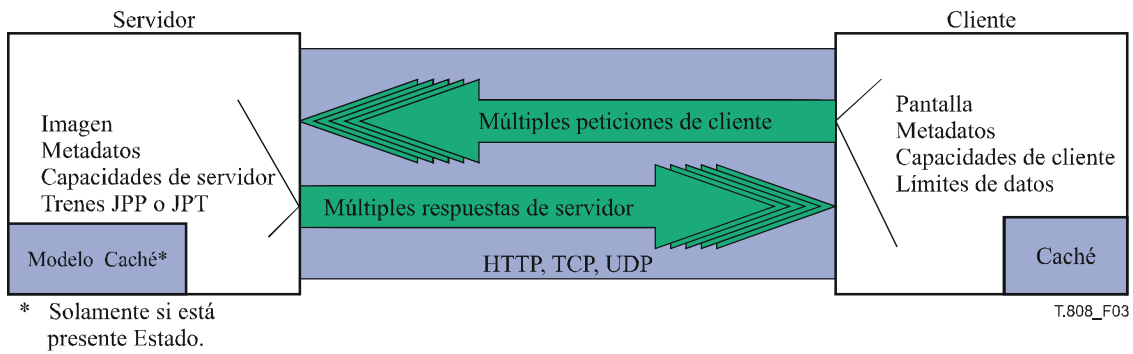


Figura 3 – Visión general del protocolo JPIP

Este protocolo se puede utilizar sobre diferentes capas de transporte, como se muestra en la figura 4. En esta Recomendación | Norma Internacional se incluyen anexos informativos acerca de la utilización del protocolo JPIP con HTTP y TCP, y se hacen sugerencias para el caso de otras implementaciones.

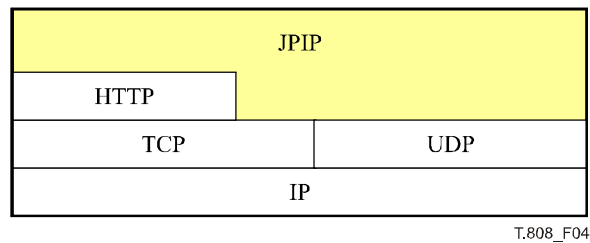


Figura 4 – Pila de protocolo JPIP

Se ha tenido en cuenta la posible extensión del protocolo JPIP para que soporte las normas actuales JPEG 2000 Rec. UIT-T T.802 | ISO/CEI 15444-3, JPEG 2000 para imágenes en movimiento, e ISO/CEI 15444-6, documentos compuestos, y las partes futuras de JPEG 2000 (actualmente JP3D, JPSEC y JPWL).

6.2 Objetivo

En esta Recomendación | Norma Internacional se definen la sintaxis y métodos que han de utilizar tanto el cliente como el servidor. En cada anexo se define un componente necesario para lograr el interfuncionamiento y la funcionalidad entre el cliente y servidor a través de varias capas de transporte. Es posible que cada anexo constituya un requisito de cliente, de servidor o de ambos.

- En el anexo A se describen los trenes basados en losas y recintos que deben utilizar los dos. El servidor ha de producir trenes JPP y JPT conformes y reconocer trenes JPP y JPT recibidos. Tras recibir una imagen parcial del servidor, el cliente ha de poder reconocer y decodificar adecuadamente dichos trenes y debe producir trenes conformes a la norma.
- En el anexo B se describe el modelo de sesión y caché de una sesión cliente/servidor, tanto para el cliente como para el servidor.
- En el anexo C se define la sintaxis de petición de cliente. Mientras que el cliente debe emitir peticiones conformes a la norma, el servidor debe poder recibir y responder a todas aquellas que sean conformes.
- En el anexo D se define la sintaxis de respuesta de servidor. El servidor debe emitir respuestas conformes a la norma y el cliente ha de poder recibir respuestas conformes.
- En el anexo E se define la sintaxis y los métodos para los sistemas que utilizan JPIP al cargar una imagen parcial.
- En los anexos F, G y H se definen los métodos y procedimientos para las interacciones cliente/servidor JPIP entre diferentes protocolos de transporte.

- En el anexo I se define la sintaxis de información de indexación contenida en una casilla JPEG 2000 y que tanto cliente como servidor pueden utilizar para acceder más eficazmente imágenes y datos relativos a ellas.
- En el anexo J se señala cómo se puede extender esta norma mediante el registro.
- En el anexo K se describen varios ejemplos de utilización de esta Recomendación | Norma Internacional para varias aplicaciones diferentes.

7 Conformidad

Un cliente se conforma a esta Recomendación | Norma Internacional siempre que sus peticiones JPIP estén bien estructuradas, sean válidas y se ajusten a las peticiones de cliente JPIP, tal como se define en la presente Recomendación | Norma Internacional. Los clientes han de soportar todos los tipos normativos de petición.

Un servidor será conforme a esta Recomendación | Norma Internacional siempre que sus respuestas JPIP estén bien estructuradas, sean válidas y conformes a las señalización de respuesta del servidor JPIP, tal como se define en la presente Recomendación | Norma Internacional. Los servidores deben soportar todos los tipos de petición normativos.

Si bien cabe esperar que esta Recomendación | Norma Internacional se implemente de tal manera que se soliciten los datos de imagen en peticiones JPIP eficientes, basándose en requisitos de aplicación del lado de cliente, no se define un comportamiento conforme.

Del mismo modo, los datos de imagen se han de ofrecer sobre la base de respuestas eficaces de servidor JPIP, minimizando así la cantidad de datos ofrecidos que no corresponden al interés del cliente y que son redundantes al estar ya en posesión de éste. No obstante, no se define un comportamiento conforme.

Cabe esperar que se pueda reducir la eficiencia de las aplicaciones de servidor mediante el envío de datos adicionales o redundantes, dependiendo del tipo de calidad de servicio de red. Estas decisiones relativas a la implementación dependen de la aplicación y permiten que el sistema JPIP sea bastante útil. No obstante, en esta Recomendación | Norma Internacional no se define la conformidad de este tipo de decisiones de implementación.

Anexo A

Los tipos de medios tren JPP y tren JPT

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

A.1 Introducción

Los trenes JPP y JPT son tipos de medios bastante útiles para presentar datos de trenes de código y formatos de ficheros JPEG 2000 en orden arbitrario. Cada tipo de medio está compuesto por una secuencia concatenada de mensajes, en la que cada uno contiene una porción de un único bin de datos precedido por un encabezamiento de mensaje. Los bins de datos contienen porciones de una representación de imagen comprimida JPEG 2000, de tal manera que es posible construir un tren que represente completamente la información presente en un fichero o tren de código JPEG 2000. Cada mensaje puede autodescribirse completamente, por lo que es posible terminar la secuencia en cualquier lugar y reordenar los mensajes con restricciones mínimas, sin que pierdan su significado. Siendo así, los tipos de medios tren JPP y tren JPT son útiles para los servidores JPIP y, por ende, el protocolo JPIP se diseña pensando particularmente en ellos. En este anexo se definen los tipos de medios tren JPP y tren JPT sin hacer referencia al protocolo JPIP.

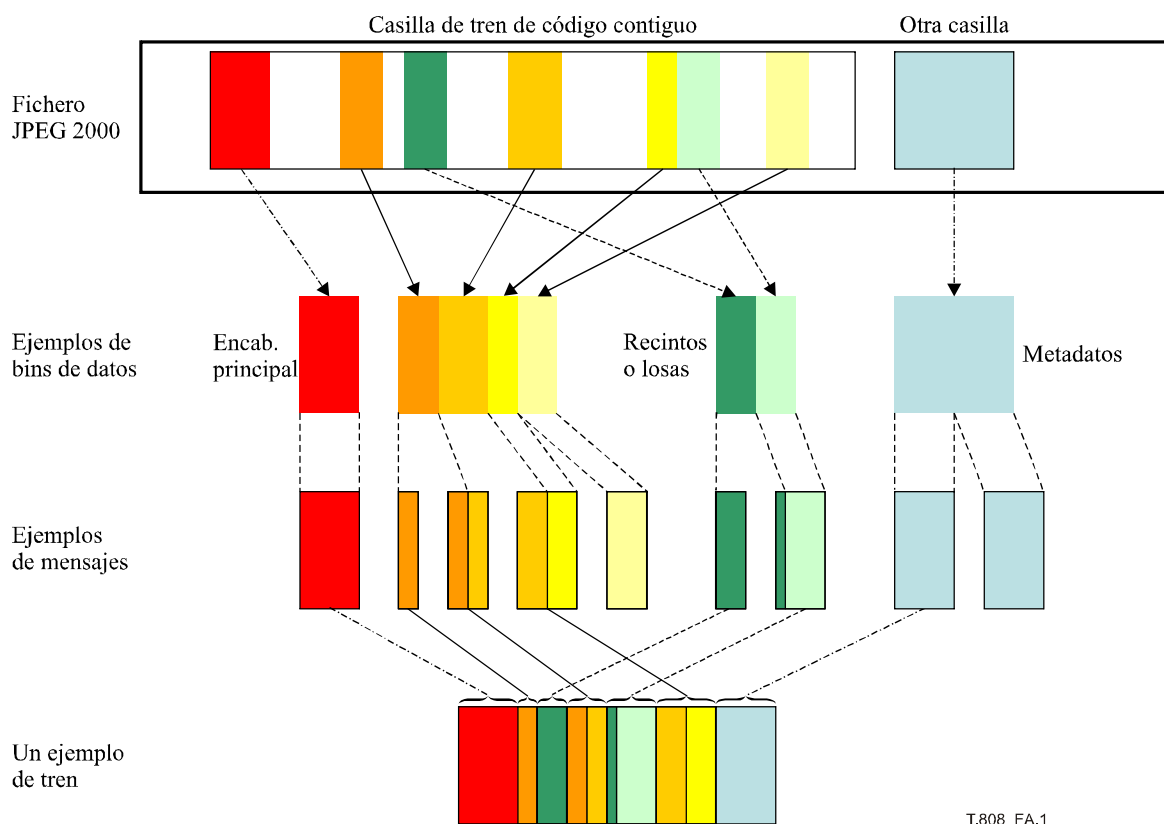


Figura A.1 – Ejemplos de fichero JPEG 2000, bins de datos JPIP y relaciones de tren JPIP (tomado de G.J. Colyer and R.A. Clark, IEEE Trans. Consumer Electronics, 49 (2003), pp 850–854)

La figura A.1 es un ejemplo bastante ilustrativo de la relación entre los trenes de bits de un fichero JPEG 2000, los bins de datos JPIP y un tren JPIP. Se muestra allí el encabezamiento principal con color rojo, 2 recintos con paquetes codificados utilizando sombras de naranja-amarillo y verde, y una casilla de metadatos en azul. A partir de estos bins de datos se pueden formar mensajes JPIP autodescritos y concatenarlos para formar un tren JPIP.

Un tren JPIP está compuesto por uno o varios mensajes JPIP concatenados, cada uno de los cuales consta de un encabezamiento y un cuerpo. En el encabezamiento se suministra información descriptiva para identificar los bins de datos pertinentes, y en el cuerpo se encuentran los bins de datos propiamente dichos. A menos que se indique lo contrario, el mensaje está compuesto por la concatenación del encabezamiento con el cuerpo.

NOTA – En esta Recomendación | Norma Internacional todos los ejemplos se refieren a mensajes binarios formados por concatenación de encabezamiento y cuerpo. Depende del tipo de transporte e implementación si se proporciona otra opción. Por ejemplo, se puede implementar la señalización auxiliar con protección de error variable en aplicaciones inalámbricas.

A.2 Estructura de encabezamiento de mensaje

A.2.1 Generalidades

Cada mensaje representa una porción de exactamente un bin de datos. El encabezamiento de mensaje está conformado por una secuencia de segmentos alineados por byte de longitud variable (VBAS, *variable-length byte-aligned segments*). Cada VBAS contiene una secuencia de bytes, todos los cuales, salvo el último, tienen su bit más significativo (bit 7) igual a 1, como se muestra en la figura A.2. Se concatenan los 7 bits menos significativos de cada byte en la VBAS para formar así un tren de bits que puede ser utilizado de diferentes maneras para diversos VBAS.

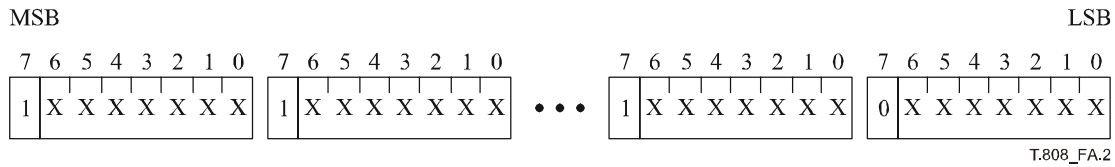


Figura A.2 – Estructura de VBAS

Gracias al encabezamiento de mensaje, se puede identificar el bin de datos y el intervalo de bytes representados por el cuerpo de mensaje. Los encabezamientos de mensaje pueden ser de forma independiente o dependiente, la primera de ellas es una forma larga cuyos encabezamientos son completamente autodescritos; su implementación es independiente de cualesquiera otros encabezamientos de mensaje. La forma dependiente, más corta, que es facultativa, utiliza información de los encabezamientos de los mensajes anteriores y tiene una decodificación que depende del inmediatamente anterior. Las aplicaciones pueden decidir utilizar la forma larga de encabezamientos de mensaje y los mensajes se pueden reordenar de manera arbitraria. De otra parte, las aplicaciones pueden usar la forma más corta de encabezamiento de mensaje que es independiente de los encabezamientos previos de mensaje y, aunque se trata de mensajes más cortos, puede ocurrir que se creen resultados con error si no se ordenan los mensajes en la secuencia correcta tras decodificarlos. Depende de la aplicación que se pueda suponer que la secuencia de ordenamiento de mensajes recibidos es fiable y, de serlo, si es posible utilizar la forma más corta de encabezamientos de mensaje.

El encabezamiento de mensaje está formado por los siguientes VBAS (se muestran los VBAS facultativos entre corchetes):

Bin-ID [, Class] [, CSn], Msg-Offset, Msg-Length [, Aux]

Al estudiar el VBAS de Bin-ID se puede determinar si hay los VBAS Class y CSn. Se puede establecer si hay un VBAS Aux mediante el VBAS Class o, de no haberlo en el encabezamiento actual de mensaje, utilizando el anterior VBAS Class.

El VBAS Bin-ID tiene varias funciones, a saber, los bits 6 y 5 del primer byte del VBAS Bin-ID, denominados 'b' en la figura A.3, indican si están presentes los VBAS Class y CSn en el encabezamiento de mensajes. En el cuadro A.1 se definen los valores de bits y su significado.

El bit 4 del primer byte del VBAS Bin-ID, denominado 'c' en la figura A.3, indica si el mensaje contiene o no el último byte en el bin de datos asociado: '0' indica que no es el último byte en el bin de datos mientras que '1' indica que sí lo es. La recepción de un mensaje que tenga este bit permite establecer la longitud de todo el bin de datos, aunque no implica que el tren JPP o el tren JPT completo tenga suficientes mensajes como para ensamblar todos los bytes de dicho bin de datos.

Los 4 bits restantes del primer byte y los 7 bits de orden inferior de cualquier otro byte en el VBAS Bin-ID (denominados 'd' en la figura A.3) forman un "identificador en clase", que se usa para identificar unívocamente el bin de datos dentro de su clase, tal como se describe en A.2.3.

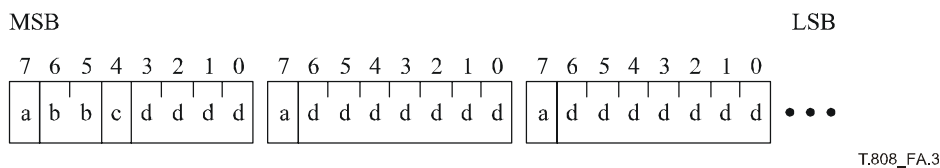


Figura A.3 – Estructura del VBAS Bin-ID

Cuadro A.1 – Indicación de VBAS adicional de Bin-ID

Indicador Bits "bb"	Significado
00	Prohibido
01	No hay VBAS Class o CSn en el encabezamiento de mensaje
10	Hay VBAS Class pero no CSn en el encabezamiento de mensaje
11	Ambos VBAS Class y CSn están presentes en el encabezamiento de mensaje

De haberlo, el VBAS Class proporciona un identificador de clase de mensaje, que es un entero no negativo formado mediante la concatenación de los 7 bits menos significativos de cada byte del VBAS en orden big-endian. Si no hay VBAS Class, el identificador de clase de mensaje es el mismo asociado con el mensaje anterior. Si no hay VBAS Class ni mensaje anterior, el identificador de mensaje es 0. En A.2.2 se describen los identificadores válidos de clase de mensaje.

De haberlo, el VBAS CSn identifica el índice (empezando desde 0) del tren de código al cual pertenece el bin de datos. El índice de tren de código se forma mediante la concatenación de los 7 bits menos significativos de cada byte del VBAS en orden big-endian. Si no hay VBAS CSn, el índice del tren de código es el mismo del mensaje anterior, y cuando no haya mensaje anterior, se pone a 0.

Los VBAS Msg-Offset y Msg-Length representan cada uno valores enteros no negativos, que se forman mediante la concatenación de los 7 bits menos significativos de cada byte en el VBAS en orden big-endian. El entero Msg-Offset indica la traslación de los datos en este mensaje a partir del inicio del bin de datos. El Msg-Length identifica la cantidad total de bytes en el cuerpo del mensaje.

Puede haber un VBAS Aux. Su presencia, y significado si está presente, vienen dados por el identificador de clase de mensaje encontrado dentro del VBAS Bin-ID, como se explica en A.2.2. Cuando hay un VBAS Aux, éste indica un valor entero no negativo, formado a través de la concatenación de los 7 bits menos significativos de cada byte en el VBAS en el orden big-endian.

NOTA – La información presente en el VBAS Aux no puede influir en la longitud del cuerpo de mensaje.

A.2.2 Identificadores de clase de mensaje

Los identificadores de clase de mensaje definidos en la presente Recomendación | Norma Internacional son enteros no negativos, como se muestra en el cuadro A.2. En A.3 se describe la interpretación de las clase de bins de datos correspondientes. Se reservan todos los demás valores de los identificadores de clase de mensaje y los mensajes que estén asociados a ellos han de ser ignorados por los decodificadores que no reconozcan su valor.

Se escogen los identificadores de clase de tal manera que haya un VBAS Aux solamente cuando el identificador sea impar, permitiendo así que se analice correctamente la sintaxis de los encabezamientos de mensajes no reconocidos y que los contenidos se descarten.

Los mensajes de bins de datos de recinto ampliados se interpretan exactamente del mismo modo que los de recinto no ampliados y se refieren a los mismos bins de datos de recinto. Los mensajes de recinto ampliados incluyen un VBAS Aux que identifica la cantidad de paquetes completos (capas de calidad) de que se dispondría si los bytes en este mensaje fueran combinados con todos los bytes anteriores del mismo recinto. Cuando este mensaje contenga también el último byte del bin de datos, el VBAS Aux indica la cantidad total de capas de calidad asociadas con el recinto en el tren de código original. De lo contrario, indica la capa de calidad a la que pertenece el byte que viene inmediatamente después del último byte en el mensaje. Algunos clientes pueden encontrar útil la información incluida en el VBAS Aux.

Cuadro A.2 – Identificadores de clase para las diversas clases de mensajes de bin de datos

Identificador de clase	Clase de mensaje	Clase de bin de datos	Tipo de tren
0	Mensaje de bin de datos de recinto	Bin de datos de recinto	Solamente tren JPP
1	Mensaje de bin de datos de recinto ampliado	Bin de datos de recinto	Solamente tren JPP
2	Mensaje de bin de datos de encabezamiento de losa	Bin de datos de encabezamiento de losa	Solamente tren JPP
4	Mensaje de bin de datos de losa	Bin de datos de losa	Solamente tren JPT
5	Mensaje de bin de datos de losa ampliado	Bin de datos de losa	Solamente tren JPT
6	Mensaje de bin de datos de encabezamiento principal	Bin de datos de encabezamiento principal	Tren JPP y tren JPT
8	Mensaje de bin de metadatos	Bin de metadatos	Tren JPP y tren JPT

Los mensajes de bins de datos de losa ampliados se interpretan exactamente de la misma manera que los de losa no ampliados y se refieren a los mismos bins de datos de losa. Los mensajes de losa ampliados incluyen un VBAS Aux que identifica el menor entero n tal que, para todos los componentes cuyo $(N_L - n)$ no es negativo, el nivel de resolución $(N_L - n)$ y los niveles por debajo de él se han completado al combinar los bytes en este mensaje con los bytes precedentes en la misma losa, y donde N_L es la cantidad de niveles de descomposición, que puede variar entre componentes. Si no se ha completado ningún nivel de resolución o componente, el valor del VBAS Aux es uno más el valor máximo de N_L en todas las componentes. Se llega al valor cero cuando hayan sido completadas todas las resoluciones en todas las componentes. Como las resoluciones no necesariamente aparecen en orden en una losa, puede ocurrir que se completen algunos niveles de resolución por encima del valor señalado por el VBAS, aunque el encabezamiento del mensaje no pueda determinarlo. Algunos usuarios pueden encontrar útil la información incluida en el VBAS Aux.

A.2.3 Identificadores en clase

Los 4 bits menos significativos del primer byte y los 7 bits menos significativos de todos los otros bytes del VBAS Bin-ID se concatenan en orden big-endian para formar una sola palabra, que tiene $7k-3$ bits, donde k es el número de bytes en el VBAS. Esta palabra representa un entero sin signo, útil para identificar unívocamente el bin de datos dentro de su clase y tren de códigos. En A.3 se suministra una descripción de las diversas clases de bins de datos, junto con los identificadores en clase correspondientes.

A.3 Bins de datos

A.3.1 Introducción

Los bins de datos contienen porciones de datos de ficheros o trenes de código JPEG 2000, y pueden basarse en elementos de imagen, tales como datos basados en el recinto, datos basados en losas y encabezamientos. También se pueden basar en metadatos. Independientemente de su contenido, cada bin de datos se trata como un solo tren de bits.

A.3.2 Bins de datos de recinto

A.3.2.1 Formato de bins de datos de recinto

Los bins de datos de recinto existen solamente en el tipo de medios de tren JPP. Cada bin de datos de recinto corresponde a un solo recinto dentro de un solo tren de código. El identificador en clase se define mediante la ecuación A-1.

$$I = t + (c + s \times \text{num_components}) \times \text{num_tiles} \quad (\text{A-1})$$

donde:

- I es el identificador único del recinto dentro de su tren de código;
- t es el índice (empezando desde 0) de la losa a la cual pertenece el recinto;
- c es el índice (empezando desde 0) de la componente de imagen a la cual pertenece el recinto;
- s es un número de secuencia que identifica el recinto dentro de su componente losa.

Dentro de cada componente losa, se atribuyen a los recintos una secuencia de números contiguos, s , como se explica a continuación. Se empieza por todos los recintos del menor nivel de resolución, (aquellos que contienen solamente muestras de subbanda LL), empezando desde el 0, y siguiendo un orden de izquierda a derecha y de arriba a abajo. A su vez, se secuencian los recintos de cada nivel de resolución sucesivo, siguiendo de nuevo un orden de izquierda a derecha y de arriba a abajo dentro de su propio nivel de resolución.

Siendo así, un identificador de recinto 0 se refiere al recinto más arriba y a la izquierda de la subbanda LL de la componente de imagen 0 en la losa 0.

Cada bin de datos de recinto corresponde a la cadena de bytes formada al concatenar todos los paquetes de tren de código, incluidos todos los encabezamientos de paquete pertinentes, que pertenecen al recinto. Cabe esperar que se puedan empaquetar los encabezamientos de paquete en segmentos de marcador PPM o PPT, que a su vez tendrán que pertenecer a los bins de datos del encabezamiento principal o de losa, en cuyo caso los bins de datos de recinto podrán tener solamente cuerpos de paquete. En todo caso, el tren de datos de recinto ha de coincidir con el segmento de bytes contiguo que habría en un tren de código JPEG 2000 que tuviese una de las secuencias de progresión subordinada a capa (CPRL, PCRL o RPCL).

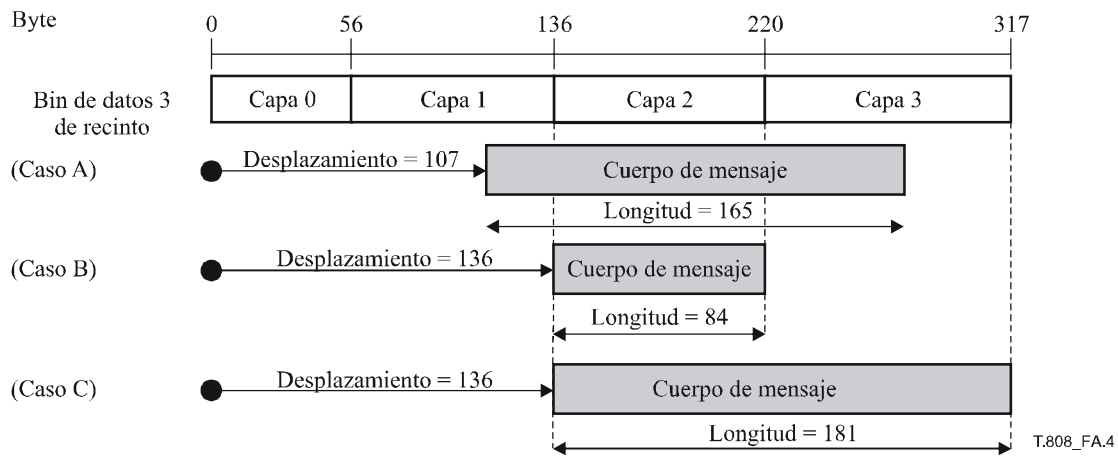


Figura A.4 – Ejemplo de bin de datos de recinto

A.3.2.2 Ejemplo de bin de datos de recinto (informativo)

En la figura A.4 se muestra un ejemplo de bin de datos de recinto (con identificador 3 en clase) que contiene 4 capas de calidad (o paquetes).

Para los casos A, B y C, se muestra a continuación el encabezamiento de mensaje, basándose en las estructuras de mensaje de bins de datos de recinto ampliado y no ampliado. Los datos que se subrayan indican el VBAS Aux para identificar el número de capas completadas por el mensaje.

(Caso A)

Encabezamiento no ampliado: 00100011 01101011 10000001 00100101 xxxxxxxx ...

El bit 0 inicial indica que se utiliza solamente un byte en el VBAS Bin-ID. Los dos bits siguientes ("01") señalan que no hay VBAS Class o CSn. El "0" siguiente quiere decir que el bin de datos no ha sido completado por este mensaje. Los demás bits del primer byte ("0011") indican que el Bin-ID es el número 3. El primer bit del segundo byte señala que se utiliza solamente un byte en el VBAS Msg-Offset. Los 7 bits siguientes ("1101011") significan que el desplazamiento es 107. El primer bit del tercer byte implica que tanto este byte como por lo menos el próximo byte forman parte del VBAS Msg-Length. El bit 0 que va al inicio del 4º byte señala que es el último byte del VBAS Msg-Length. Todos los bits de orden inferior del 3º al 4º byte, por tanto, concatenados para establecer la longitud. En este caso, "0000001 0100101" = 165.

Encabezamiento ampliado: 01000011 00000001 01101011 10000001 00100101 00000011 xxxxxxxx ...

(Caso B)

Encabezamiento no ampliado: 00100011 10000001 00001000 01010100 xxxxxxxx ...

Encabezamiento ampliado: 01000011 00000001 10000001 00001000 01010100 00000011 xxxxxxxx ...

(Caso C)

Encabezamiento no ampliado: 00110011 10000001 00001000 10000001 00110101 xxxxxxxx ...

Encabezamiento ampliado: 01010011 00000001 10000001 00001000 10000001 00110101 00000100 xxxxxxxx ...

Cabe observar que al contener los datos de respuesta el último byte del bin de datos en el caso C, el VBAS Bin-ID indica que se trata de un mensaje que ha sido "completado".

A.3.3 Bins de datos de encabezamiento de losa

Los bins de datos de encabezamiento de losa están presentes solamente dentro del tipo de medios tren JPP. Cuando los bins de datos pertenezcan a esta clase, el identificador en clase mantiene el índice (empezando desde 0) de la losa a la que se refiere el bin de datos. Dicho bin de datos está compuesto por marcadores y segmentos de marcador para la losa n, y no podrá incluir un segmento de marcador SOT. Se puede también incluir marcadores SOD. Es posible formar este bin de datos a partir de un tren de código válido, concatenando todo los segmentos marcadores salvo SOT y POC en todos los encabezamientos de parte losa para la losa n.

A.3.4 Bins de datos de losa

Los bins de datos de losa se han de utilizar solamente con el tipo de medios tren JPT. Para los bins de datos de esta clase, el identificador en clase es el índice (empezando desde el número 0) de la losa a la que pertenece el bin de datos. Cada bin de datos de losa corresponde a la cadena de bytes formada al concatenar todas las partes losa que pertenecen a la losa, en orden, completada con sus SOT, SOD y todos los otros segmentos marcadores del caso.

A.3.5 Bin de datos de encabezamiento principal

Tanto los tipos de medios de tren JPP como JPT utilizan el bin de datos de encabezamiento principal. Para los bins de datos que pertenecen a la clase de encabezamiento principal de tren de código (variaciones completada o no completada) el identificador en clase será 0. Este bin de datos está compuesto por una lista concatenada de todos los marcadores y segmentos de marcador en el encabezamiento principal, empezando desde el marcador SOC. No contiene marcadores SOT, SOD o EOC.

A.3.6 Bins de metadatos

A.3.6.1 Introducción a los bins de metadatos

Ambos tipos de medios, el tren JPP y el JPT, utilizan los bins de metadatos para transportar metadatos desde el objetivo lógico que contiene el tren o los trenes de códigos cuyos elementos pueden ser referenciados por otros bins de datos asociados con el tren JPP o JPT. A efectos de esta Recomendación | Norma Internacional, el término "metadatos" se refiere a cualquier colección de "casillas" de un fichero de la familia JPEG 2000. Se ignorará el índice del tren de códigos en todo mensaje que contenga el identificador de clase de bin de metadatos.

A diferencia de los ID numéricos que se utilizan para otros tipos de bins de datos, el ID de bin de metadatos no corresponde algorítmicamente con algún formato de fichero o desplazamiento de byte. Siendo así, el servidor puede escoger cualquier ID numérico para determinado bin de metadatos. La única excepción consiste en que el bin de metadatos que contenga la raíz del objetivo lógico habrá de tener como ID 0.

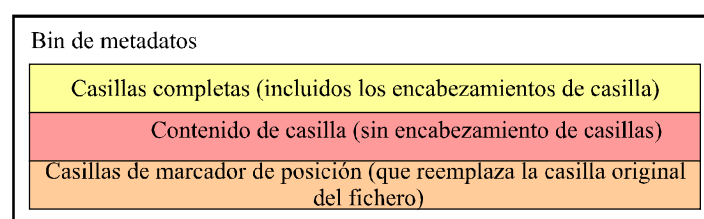
NOTA – Si bien el mecanismo de atribución depende de la implementación, se sugiere informativamente que los servidores atribuyan los ID de bin utilizando números consecutivos.

A.3.6.2 División en bins de metadatos de un objetivo lógico que contenga un fichero JPEG 2000

Es concebible imaginar que todos los metadatos se puedan incluir en el bin de metadatos 0. En este caso, todas las casillas del objetivo lógico pertenecen al bin de metadatos 0, y figuran en su orden original. Al estar compuestos los formatos de ficheros de la familia JPEG 2000 por secuencias de casillas, el bin de metadatos 0 habría de incluir todo el objetivo lógico. Sin embargo, es útil dividir el objetivo lógico en piezas que se puedan transmitir de una manera manejable, permitiendo a los servidores de imagen omitir deliberadamente porciones del objetivo lógico que no necesite el cliente en ese momento. A este fin, en JPIP se define un nuevo tipo de casilla especial, denominada la "casilla marcador de posición". Esta casilla permite identificar el tamaño y tipo de una casilla del objetivo lógico, al tiempo que apunta a otro bin de datos que tiene los contenidos de la casilla. Los marcadores de posición también pueden representar trenes de código del objetivo lógico, algo que reviste particular importancia si se tiene en cuenta el hecho de que los datos comprimidos representados por determinado tren de código pueden ser entregados incrementalmente a través de otros tipos de bins de datos (bins de datos de encabezamiento y bins de datos de recinto, o bins de datos de losa).

El bin de metadatos 0 está compuesto, en la debida forma, por todas las casillas del objetivo lógico, en su orden original, salvo que cualquiera de ellas puede ser reemplazada por un marcador de posición. La casilla marcador de posición incluye el encabezamiento original de la casilla que ha sido reemplazada, junto con el identificador del bin de metadatos que mantiene los contenidos de la casilla, sin incluir el encabezamiento en sí. Todos los bins de metadatos diferentes del 0 estarán compuesto por el contenido de alguna casilla cuyo encabezamiento aparezca en el marcador de posición que haga referencia a dicho bin de datos. A su vez, dichos contenidos de casilla pueden incluir subcasillas, cualquiera de las cuales se puede reemplazar utilizando otros marcadores de posición.

Se utilizará la siguiente notación de colores para ilustrar los ejemplos de bins de metadatos (véase la figura A.5):



T.808_FA.5

Figura A.5 – Ejemplo de notación de colores para bins de metadatos

Considérese, por ejemplo, un fichero JP2 que tenga la siguiente estructura de casillas (figura A.6):

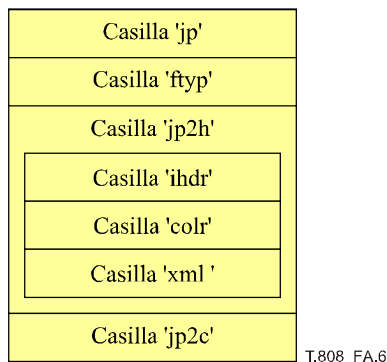


Figura A.6 – Muestra de fichero JP2

Este fichero se puede dividir en tres bins de metadatos, a saber, uno que representa el nivel superior del fichero original (bin de datos 0), uno que representa la casilla de encabezamiento JP2 y otro que representa el tren de código. En la figura A.7 se muestra esa división.

Mientras que el contenido de cualquier bin de metadatos tiene que ser el contenido de la casilla o fichero representado por él, los datos reales allí incluidos pueden variar conceptualmente dependiendo del tipo de casilla. Así, por ejemplo, en el bin de metadatos 1 de la figura A.7, que representa el contenido de la casilla de encabezamiento JP2, el contenido de dicha casilla es literalmente una serie de otras casillas completas, como si el encabezamiento JP2 fuese una supercasilla. Al no haber otros datos en la casilla de encabezamiento JP2, no habrá dentro del bin de metadatos 1 ningún otro dato que forme parte de la serie de aquellas casillas completas. Los datos dentro del bin de metadatos 2 son, por el contrario, el contenido bruto de la casilla de tren de código contiguo, sin encabezamientos de casilla, pues ésta es una supercasilla.

Cabe observar que en el ejemplo de la figura A.7 el acceso a los datos de tren de código se puede efectuar de dos maneras diferentes. Se utiliza el segundo bin marcador de posición para reemplazar la casilla de tren de código contiguo (jp2c) en el fichero original, identificando así el bin de metadatos 2 como el que tiene el contenido original de dicha casilla, es decir, el tren de código bruto propiamente dicho. A efectos de descripción de la presente Recomendación | Norma Internacional, esto se denominará la representación de "tren de código bruto". Los trenes de código brutos se ofrecen a partir de los bins de metadatos.

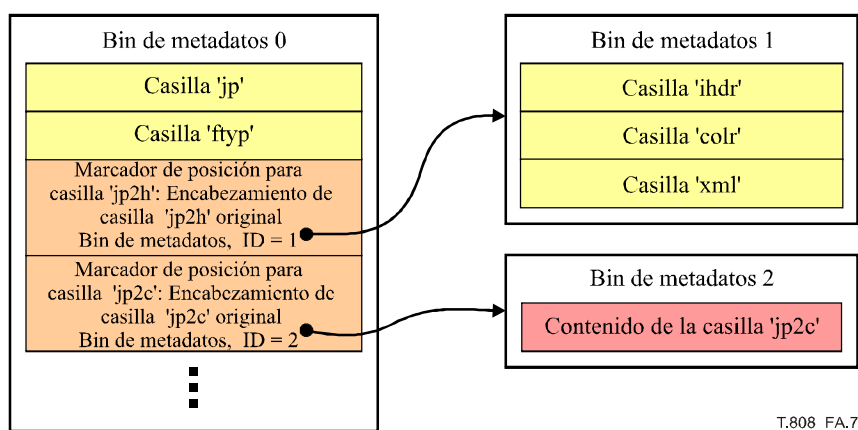


Figura A.7 – Muestra de fichero JP2 dividido en tres bins de metadatos

El marcador de posición puede también suministrar un identificador de tren de código. Todo bin de datos que pertenezca a las clases encabezamiento principal, encabezamiento de losa, bin de datos de recinto o losa, y que tenga el mismo identificador de tren de código, transporta datos comprimidos relacionados con el mismo tren de código encontrado en el bin de metadatos 2. A efectos de la descripción de la presente Recomendación | Norma Internacional, esto se denominará como representación "tren de código incremental", que se ofrecerán a partir de estos bins de datos.

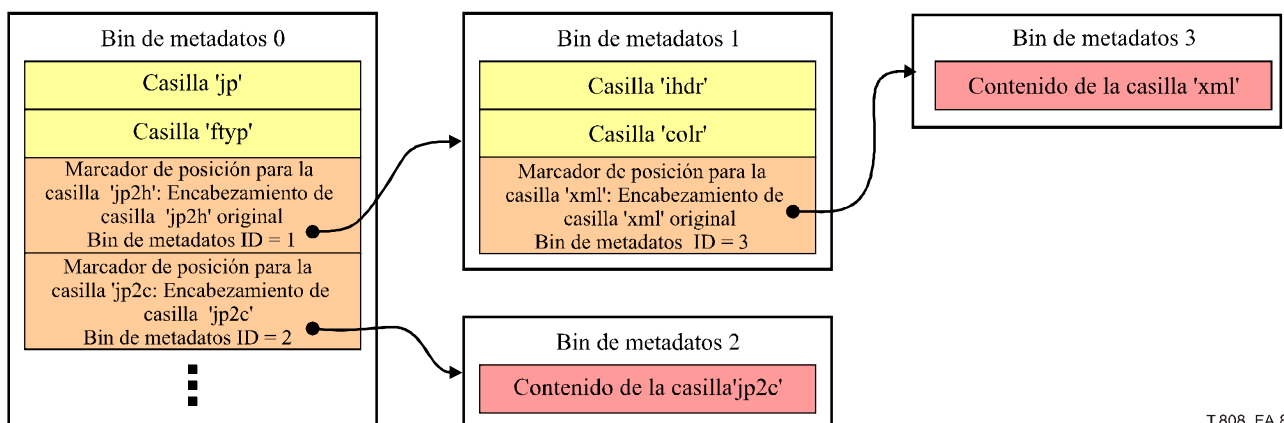
En general, los marcadores de posición que referencian tren de código pueden hacerlo ya sea mediante un bin de metadatos separado (tren de código bruto) o proporcionando un identificador de tren de código (tren de código incremental), o ambos. Aun si existen ambos métodos, es posible que los datos de tren JPP o JPT de que dispone un cliente o un agente que produce imágenes tengan solamente el contenido del tren de código bruto o solamente datos del tren de código incremental. Más aún, aunque se disponga tanto de las versiones bruta como de la incremental del mismo tren de códigos, no se puede garantizar que las dos representaciones tengan parámetros de codificación compatibles. Sólo se podrá garantizar la coherencia de las muestras de imagen reconstruidas asociadas con las dos representaciones.

Se puede también utilizar casillas de marcador de posición para asociar múltiples trenes de códigos con una sola casilla original, en cuyo caso la interpretación depende de la casilla que se haya reemplazado. En A.3.6.4 se discute con más detalle este asunto.

En el ejemplo simple de la figura A.7, las casillas de marcador de posición aparecen solamente en el nivel superior del fichero, en el bin de metadatos 0. No obstante, como ya se ha señalado, también, se pueden utilizar los marcadores de posición para reemplazar cualquier casilla en cualquier bin de metadatos. De esta manera es posible descomponer jerárquicamente ficheros complejos, por lo que sólo el fichero original podrá ser encapsulado en varias estructuras diferentes de bin de metadatos, según como se utilicen los marcadores de posición. **Sin embargo, un solo tren JPP o JPT habrá de utilizar solamente una de dichas encapsulaciones.** En las aplicaciones cliente-servidor, suele ocurrir que el servidor determine una estructura adecuada de bin de metadatos para el fichero, atribuyéndole un identificador único al tren resultante y utilizando la misma estructura de bin de metadatos en todas las comunicaciones con todos los clientes relacionadas con este identificador único.

Siempre que un marcador de posición reubique una casilla en un nuevo bin de metadatos, el encabezamiento de aquella (campos LBox, TBox y XLBox) se almacena sin modificación en la casilla marcador de posición. Si un cliente o agente que produce imágenes necesita hacer corresponder determinadas casillas con sus desplazamientos originales de fichero, podrá hacerlo utilizando los encabezamientos originales de casilla que figuran en las casillas marcador de posición. En últimas, esta información permite que se haga corresponder cualquier ubicación del fichero original a cualquier otra en determinado bin de metadatos, siempre que existan los contenidos de dicho bin de datos. Esto es importante puesto que algunos ficheros de la familia JPEG 2000 contienen casillas que hacen referencia a otras casillas a través de su ubicación dentro del fichero.

Si bien hay bastante libertad en lo que respecta a cómo dividir un fichero en bins de metadatos, existe una restricción, a saber que toda casilla marcador de posición que figure en un bin de metadatos habrá de reemplazar una casilla de nivel superior dentro de dicho bin de datos, o lo que es lo mismo, siempre que se deba reemplazar una subcasilla con un marcador de posición, la supercasilla inmediatamente superior que la que contiene habrá de estar dentro de su propio bin de metadatos. Por ejemplo, en el fichero de la figura A.6, los datos XML incluidos dentro de la casilla de encabezamiento JP2 pueden ir en un bin de datos aparte de las otras casillas. Así, un servidor puede entregar solamente aquellos bins de datos que son necesarios en su momento para la decodificación y presentación de la imagen, a menos que se soliciten explícitamente datos XML. En la figura A.8 se muestra una estructura adecuada de bin de datos.



T.808_FA.8

Figura A.8 – Supercasilla con bin de metadatos referenciados

No obstante, no sería válido dejar la casilla de encabezamiento JP2 en el bin de metadatos 0, como se muestra en la figura A.9:

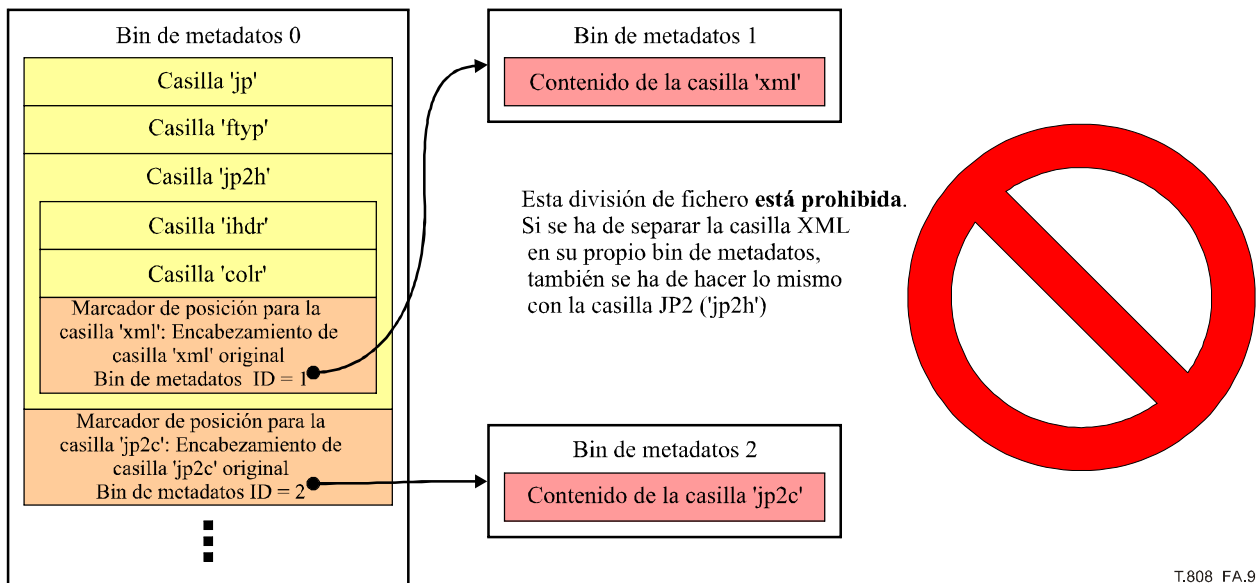


Figura A.9 – División no válida del fichero en bins de metadatos

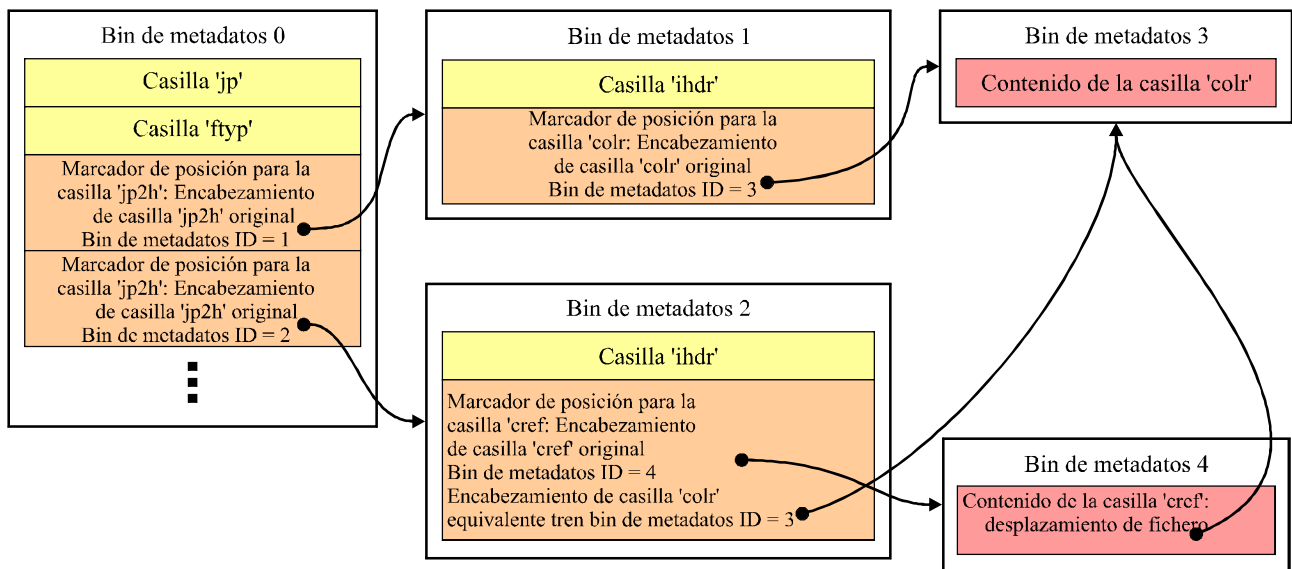
NOTA – Una forma equivalente de expresar esta misma restricción consiste en que siempre que un marcador de posición reemplace una subcasilla, otro marcador de posición tendrá que reemplazar también la casilla que la contiene. De esta manera se garantiza que un cliente o agente reproductor de imágenes pueda siempre recuperar las longitudes y ubicaciones de las casillas originales en el fichero, aun si algunas de estas casillas no han podido ser reconocidas por el cliente.

Además de suministrar los contenidos originales de una casilla en un bin de metadatos separado, los trenes JPP y JPT también pueden proporcionar otras representaciones de la casilla que no aparezcan explícitamente dentro del fichero original y que se denominan "equivalentes de tren". Por ejemplo, puede ocurrir que el fichero original contenga una casilla de referencia cruzada cuya casilla de lista de fragmentos contenga uno o varios fragmentos del fichero para reconstituir una casilla de especificación del espacio de color. Si bien un cliente o un agente reproductor de imágenes ha de poder seguir los punteros de fichero pertinentes para reconstruir la casilla de especificación del espacio de color, una representación más conveniente de tren JPP o JPT debe incluir un marcador de posición que haga referencia a un bin de datos que contenga la casilla de especificación de color reconstruida como equivalente de tren. A este fin, el marcador de posición contiene un encabezamiento de casilla para el equivalente de tren junto con el identificador del bin de metadatos que tiene el contenido de la casilla equivalente de tren.

En el ejemplo a continuación (véase la figura A.10) se indica la utilización de equivalentes de tren para las casillas de referencia cruzada. En este caso, el bin de datos que tiene el contenido de equivalente de tren se referencia también como si tuviera el contenido original de otra casilla. Aunque esto suele ocurrir cuando el fichero original contenga casilla de referencias cruzadas, no es necesario que el equivalente de tren apunte a un bin de metadatos que esté conectado a la jerarquía de fichero original. Se puede crear el contenido de las casillas de equivalente de tren desde el principio o referirlo al contenido que ya exista en otros ficheros. Siendo así, las casillas de referencia cruzadas cuya lista de fragmentos haga referencia a otros ficheros o URL han de poderse encapsular completamente dentro de un solo tren JPP o JPT.

Es posible utilizar los equivalentes de tren en cualquier situación en la que el servidor pueda crear una forma alternativa de contenido de una casilla que ofrezca algún beneficio al cliente; estos equivalentes no son útiles solamente para proporcionar acceso a datos que tienen referencia cruzada explícita.

Además de apuntar a datos de casilla reales o equivalentes, puede ocurrir que un marcador de posición apunte a uno o varios trenes de códigos cuando la casilla reemplazada sea equivalente a ellos. Por ejemplo, cabe reemplazar la casilla de tren de código contiguo por una casilla marcador de posición que haga referencia al ID del tren de código incremental contenido dentro de dicha casilla de tren de código contiguo. Otro ejemplo sería cuando se reemplaza la casilla de desplazamiento de porción (Chunk) en un fichero de movimiento JPEG 2000 por un marcador de posición que especifique un vector ID de tren código, que se refiera a los trenes de códigos a los que apunta la casilla de desplazamiento de porción.

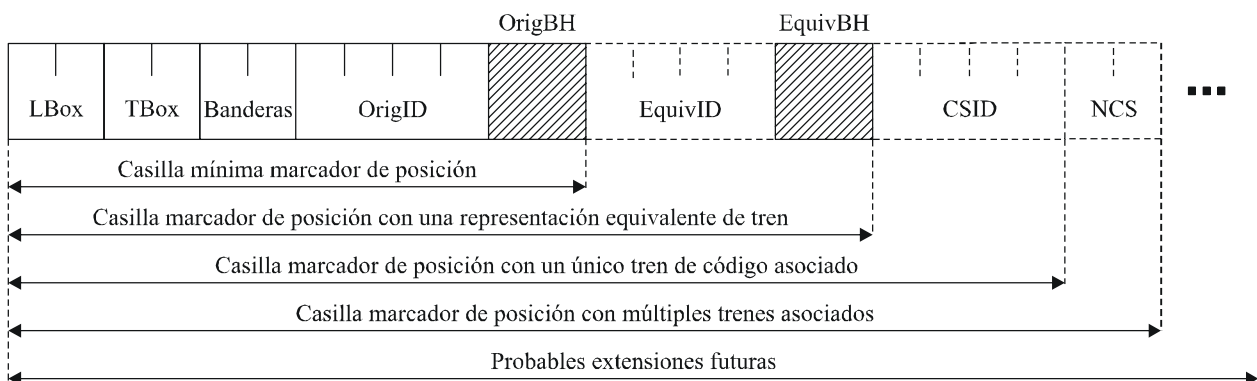


T.808_FA.10

Figura A.10 – Ejemplo de utilización de equivalentes de tren

A.3.6.3 Formato de casilla marcador de posición

En la figura A.11 se indica el formato de una casilla marcador de posición, incluido el encabezamiento de casilla (a diferencia de la definición de la mayoría de casillas en el anexo I y en otras partes de esta Recomendación | Norma Internacional); esta Recomendación | Norma Internacional tiene como objetivo hacer énfasis en que la utilización del campo longitud en el encabezamiento de casilla para el caso de una casilla marcador de posición tiene más restricciones que cuando se trata de otras casillas.



T.808_FA.11

Figura A.11 – Estructura de casilla marcador de posición

LBox: Campo de longitud big-endian estándar de 4 bytes para una casilla. En el caso de una casilla marcador de posición el valor no podrá ser 1, indicando que no puede estar presente el campo XLBox.

TBox: Campo de tipo de casilla normalizada de 4 bytes para una casilla. El valor de tipo para una casilla marcador de posición ha de ser 'phld' (0x7068 6c64).

Banderas: Campo que especifica qué elementos de la casilla marcador de posición contienen datos válidos. Se codifica como un entero big-endian de 4-bytes. Los valores válidos para el campo Banderas se especifican en el cuadro A.3.

OrigID: Campo que especifica el ID del bin de metadatos del bin que tiene el contenido de la casilla original representada por esta casilla marcador de posición. Se codifica como un entero sin signo big-endian de 8 bytes.

OrigBH: Campo que especifica el encabezamiento original (LBox, TBox y XLBox, cuando corresponda) de la casilla original referenciada por esta casilla marcador de posición. Su longitud es 8 bytes si el campo LBox del encabezamiento de casilla original no es igual a 1, y 16 bytes de lo contrario.

ISO/CEI 15444-9:2005 (S)

EquivID: Campo que especifica el ID de bin de metadatos del bin que contiene una forma equivalente de tren del contenido de esta casilla. Este campo se codifica como un entero sin signo big-endian de 8 bytes.

EquivBH: Campo que especifica el encabezamiento de la casilla equivalente de tren (LBox, TBox y XLBox, cuando corresponda) de la casilla referenciada por esta casilla marcador de posición. Su longitud es 8 bytes si el campo LBox del encabezamiento de casilla equivalente es diferente de 1 y 16 bytes de lo contrario.

CSID: Campo que especifica el ID del primer tren de código asociado con la casilla reemplazada. Éste es el ID que se asocia con todos los bins de datos de encabezamiento, recinto y/o losas que se utilizan para comunicar incrementalmente el contenido del primer tren de códigos relacionado con la casilla reemplazada. Se codifica como un entero sin signo big-endian de 8 bytes.

NCS: Campo que especifica la cantidad de trenes de código en el conjunto de trenes de código equivalente a la casilla reemplazada. Los valores del ID de tren de código de estos trenes de código van contiguamente desde el valor especificado por el campo CSID. Se codifica como un entero sin signo big-endian de 4 bytes.

ExtendedBoxList: No se muestra específicamente en la figura A.11. El campo NCS puede ir seguido de una secuencia de casillas que contengan información ampliada del servidor. La existencia de cualquier casilla tras el campo NCS ha de especificarse a través de un bit en el campo Flags. No obstante, en esta Recomendación | Norma Internacional no se definen casillas extendidas ni banderas de bit adicionales. Los clientes han de ignorar cualquier casilla en la ExtendedBoxList que no hayan reconocido.

Un valor de bit "x" en el cuadro A.3 indica que el valor especificado incluye casos en los que el bit se pone bien a "1" o bien a "0". En esta norma no se utilizan los bits indicados como "y", que los servidores han de poner en 0 y los clientes ignorar.

En cada casilla marcador de posición no es necesario que aparezcan todos los campos definidos para una de estas casillas. Tal como se sugiere mediante las flechas en la figura A.11, cuando no se proporcione ID de tren de código equivalente o incremental de casilla, se puede terminar la casilla al final del campo OrigBH. Del mismo modo, si no hay ID de tren de código incremental, es posible que la casilla termine al final del campo EquivBH, y si no hay más de un ID de tren de código incremental al final del campo CSID.

Cuadro A.3 – Valores admitidos para el campo Banderas de una casilla marcador de posición

Valor	Significado
yyyy yyyy yyyy yyyy yyyy yyyy yyyy xxx1	El acceso al contenido original de esta casilla se logra mediante el bin de metadatos especificado en el campo OrigID
yyyy yyyy yyyy yyyy yyyy yyyy yyyy xxx0	No hay acceso al contenido original de esta casilla y se ha de ignorar el valor del campo OrigID
yyyy yyyy yyyy yyyy yyyy yyyy yyyy xx1x	Se proporciona una casilla equivalente tren, cuyo contenido está en el bin de metadatos especificado por el campo EquivID
yyyy yyyy yyyy yyyy yyyy yyyy yyyy xx0x	No se proporciona casilla equivalente tren y se han de ignorar los valores de cualquier campo EquivID y EquivBH
yyyy yyyy yyyy yyyy yyyy yyyy yyyy 01xx	El acceso a la imagen representada por esta casilla se obtiene mediante un solo tren de código incremental identificado por el campo CSID. Se tiene que tratar el valor del campo NCS como si estuviese puesto a "1" sin importar su valor real
yyyy yyyy yyyy yyyy yyyy yyyy yyyy 11xx	El acceso a la imagen representada por esta casilla se obtiene mediante uno o varios trenes de código incremental, como se especifica mediante los campos CSID y NCS.
yyyy yyyy yyyy yyyy yyyy yyyy yyyy x0xx	Este marcador de posición no proporciona acceso a una imagen que representa la casilla original como tren de código incremental; se han de ignorar los campos CSID y NCS
Otros valores	Reservado para la ISO

A.3.6.4 Referencia de trenes de código incremental mediante marcadores de posición

Siempre que haya bins de datos de encabezamiento, recinto o losa, sus ID de tren de código habrán de figurar en una casilla marcador de posición dentro de un bin de datos adecuado. La única excepción a este requisito será para los trenes de código JPEG 2000 no cubiertos, que no están incorporados dentro del formato de fichero de la familia JPEG 2000.

Los valores de ID de tren de código que figuran en las casillas marcador de posición pertinentes deberán cumplir con todos los requisitos impuestos por el formato del fichero que los contiene. Por ejemplo, los ficheros JPX atribuyen formalmente un número de secuencia a cada tren de código que aparece en la parte superior del fichero, bien sea a través de una casilla de tren de código contiguo o de una casilla de cuadro de fragmentación. El primer tren de código

de nivel superior en el objetivo lógico ha de tener un ID de tren de código igual a 0, el siguiente tendrá uno igual a 1 y así sucesivamente.

Sólo se podrán utilizar marcadores de posición que hagan referencia a múltiples ID de tren de código cuando el significado de dichos trenes de código esté bien definido por el tipo de la casilla que se está reemplazando.

A.3.6.5 Utilización de casillas marcador de posición con MJ2

En esta Recomendación | Norma Internacional se definen solamente dos tipos de casillas adecuados para los marcadores de posición con ficheros de imagen en movimiento JPEG 2000 (MJ2). En particular, se puede reemplazar la casilla de desplazamiento de porción ('stco') o la de desplazamiento de gran porción ('co64') por una casilla marcador de posición que identifica múltiples ID de tren de código.

Cada pista de vídeo en un fichero MJ2 contiene exactamente una casilla de desplazamiento de porción (ya sea 'stco' o 'co64'), que, junto con la muestra para casilla de porción ('stsc'), permite identificar la ubicación de todas las casillas de tren de código contiguas que pertenecen a dicha pista de vídeo. Si se reemplaza la casilla de desplazamiento de porción por un marcador de posición que proporcione uno o varios ID de tren de código, habrá exactamente un ID de tren de código para cada casilla de tren de código contigua en la pista de vídeo. Si la casilla de entrada de muestra visual ('mjp2') identifica una cuenta de campos igual a 2, habrá entonces $2N$ ID de tren de código en el intervalo suministrado por la casilla marcador de posición, donde N es el número de muestras de vídeo (es decir, N es el número de cuadros). De lo contrario, habrá solamente N ID de trenes de código en la gama proporcionada por esa casilla. Se han de ordenar los ID de tren de código en una secuencia mediante el número de muestra (número de cuadros) y conforme al número de campo dentro de cada muestra.

NOTA – Para los ficheros MJ2 en una representación de tren JPP o tren JPT, no es necesario que el tren incluya el contenido de la casilla original de desplazamiento de porción, la muestra de casilla de porción ('stsc') o la casilla de tamaño de muestra ('stsz'). Esta información de indexación se puede regenerar, de ser necesario, si se convierte la representación de tren a un fichero MJ2.

A.4 Convenios para el análisis gramatical y la entrega de trenes JPP y JPT (informativo)

Aunque las casillas marcador de posición permiten mayor flexibilidad, crean cierta posible ambigüedad tanto para los clientes como para los servidores en lo que respecta a cómo han de analizar gramaticalmente o entregar los trenes JPP y JPT. Puede ocurrir que un servidor escoja dividir las casillas originales de un fichero de la familia JPEG 2000 en bins de metadatos utilizando cualquier estrategia, mediante la introducción de casillas marcador de posición en los puntos adecuados. El servidor habrá de actuar de manera coherente para que los bins de datos correspondientes al tren JPP o JPT tengan el mismo contenido nominal para todos los clientes que accedan al mismo objetivo lógico (tal vez calificándolos mediante un solo ID objetivo), siempre que accedan a él.

No obstante, es más significativo observar que las casillas marcador de posición permiten a los servidores construir un solo tren JPP o JPT cuyos bins de datos proporcionan múltiples representaciones alternativas del mismo contenido original. Esto puede ocurrir cuando se identifica un equivalente de tren y/o un ID de tren de código incremental dentro de un marcador de posición. En estos casos, debe haber un bin de metadatos en la casilla original al mismo tiempo que también está disponible como equivalente de tren en otro bin de metadatos y/o como tren de código incremental a través de bins de datos de encabezamiento, recinto o losa. Si bien es posible que los servidores distribuyan el contenido de todos los bins de datos que representan una casilla original, por razones de eficacia cabe esperar que distribuyan solamente la información necesaria para transportar el contenido original, a menos que se solicite explícitamente distribuir los bins de datos redundantes. Al verse confrontados a múltiples representaciones de una casilla original, los analizadores de gramática del lado del cliente de trenes JPP o JPT, pueden elegir ignorar todas salvo una de ellas. Cabe esperar que la convención de cliente tenga un efecto importante sobre la selección de bins de metadatos que hace el servidor para enviarlos al cliente.

Teniendo esto en mente, en esta Recomendación | Norma Internacional se recomiendan los siguientes convenios:

- a menos que un servidor tenga una razón para creer lo contrario, habrá de suponer que el analizador de gramática del cliente dará prioridad a la casilla equivalente de tren sobre la casilla original, si los marcadores de posición han señalado al cliente la existencia de ambos tipos de casilla;
- a menos que el servidor tenga una razón para creer lo contrario, supondrá que el analizador de gramática del cliente dará preferencia a la representación de tren de código incremental (bins de datos de encabezamiento, recinto o losa), sobre el tren de código bruto, cuando los marcadores de posición le hayan señalado la existencia de ambos tipos de casilla.

A.5 Convenios para el interfuncionamiento de trenes JPP o JPT (informativo)

En este convenio se describe el formato de fichero de intercambio para trenes JPP y JPT, denominados de ahora en adelante fichero jpp y fichero jpt, respectivamente. Dichos ficheros pueden contener los datos JPEG 2000 recibidos de

ISO/CEI 15444-9:2005 (S)

una sesión JPIP (por ejemplo, la caché del cliente), o un subconjunto de ellos. Otro cliente JPIP puede leer y utilizar este fichero puesto que los trenes JPP y JPT son tipos de medios que se autodistribuyen.

Estos ficheros se forman concatenando mensajes de tren JPT o JPP. Por ejemplo, pueden estar compuestos por la concatenación simple de todos los mensajes de este tipo recibidos por un cliente en una sola sesión o en múltiples sesiones. Algo mejor ocurrirá siempre que haya clientes que generen un tren válido JPT o JPP utilizando un solo encabezamiento de mensaje y un solo mensaje por cada bin de datos.

Se recomienda utilizar las extensiones ".jpp" y ".jpt" para estos ficheros y, cuando corresponda, que el nombre del fichero haga referencia al testigo JPIP `target` o al testigo `target-id`.

En este convenio no se especifica la implementación o estructura de la caché de un cliente. Por ejemplo, puede ocurrir que un cliente utilice una base de datos como implementación de la función caché en lugar de un sistema caché basado en fichero.

Anexo B

Sesiones, canales, modelo caché y conjuntos modelo

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

B.1 Compartición entre las peticiones dentro de una sesión y las peticiones sin estado

En el protocolo JPIP se hace una clara distinción entre dos tipos de peticiones, a saber las sin estado y las que pertenecen a una sesión.

El objetivo de las sesiones es reducir la cantidad de comunicación explícita requerida entre el cliente y el servidor. Dentro de una sesión, se espera que el servidor recuerde las capacidades y preferencias del cliente proporcionadas en peticiones anteriores, de tal manera que no sea necesario enviar esta información cada vez que se haga una petición. Aún más importante, el servidor suele mantener un registro de la información que ya ha enviado al cliente como respuesta a las peticiones anteriores, con lo cual no es necesario retransmitirla para responder a peticiones futuras. Este registro habrá de persistir durante toda la sesión. A menos que se ordene explícitamente lo contrario, es posible que el servidor suponga que el cliente almacena en caché las respuestas a todas sus peticiones dentro de una misma sesión y pueda modelar la caché del cliente enviando solamente aquellas porciones de los datos o metadatos de imagen comprimida que el cliente no posee aún en su caché.

Las peticiones sin estado no corresponden a ninguna sesión y, por ende, deben ser completamente autosuficientes. Cabe observar que el término "sin estado" se aplica solamente al servidor pero no al cliente. Como ocurre con las sesiones, el cliente suele almacenar en la caché las respuestas a las peticiones previas correspondientes al mismo objetivo lógico. Siendo así, los clientes que emiten múltiples peticiones sin estado para el mismo objetivo incluirán generalmente información acerca de su contenido en caché con cada petición para evitar la transmisión de información redundante. En otras palabras, los beneficios de las sesiones son peticiones más pequeñas, menos complejas y/o respuestas servidor menos redundantes. La ventaja de la comunicación sin estado consiste en que el servidor no necesita mantener información de estado entre las peticiones, permitiendo así que no sea necesario un solo servidor para todas las peticiones referentes a una sola imagen objetivo que emanan de un cliente único.

B.2 Canales y sesiones

Se asocian los siguientes términos a cada sesión:

- Uno o varios objetivos lógicos (que suelen ser ficheros de imagen) cuyo contenido no cambia durante la sesión.
- Un solo tipo de retorno de datos de imagen para cada objetivo lógico asociado con la sesión.
- Para cada objetivo lógico asociado con la sesión, se tendrá que mantener un modelo del contenido de la caché de cliente, siempre que el tipo de retorno de datos sea "jpp-stream" o "jpt-stream". Sin embargo, cabe observar que no es necesario que este modelo refleje perfectamente el estado real de la caché de cliente. En B.3 se describen las reglas que gobiernan el mantenimiento de los modelos caché.
- Uno o varios canales JPIP. En general, los clientes pueden abrir múltiples canales dentro de la misma sesión. Es posible asociar cada canal JPIP con un canal de transporte subyacente independiente (por ejemplo, una conexión TCP separada), aunque no sea necesariamente el caso. Gracias a los múltiples canales, los clientes pueden emitir peticiones simultáneas para múltiples regiones de la imagen, esperando que el servidor responda a ellas simultáneamente. Del mismo modo, los canales permiten la atribución inteligente de ancho de banda entre los diversos tipos de peticiones sea dentro de una sola imagen objetivo o a través de múltiples objetivos.
- Siempre que se asocien múltiples canales con el mismo objetivo lógico, el modelo caché de sesión se aplica en todos los canales. Varios clientes pueden abrir canales JPIP dentro de la misma sesión aunque esto pueda tener efectos colaterales no deseados cuando los canales se refieran al mismo objetivo lógico.

Los siguientes elementos se asocian con cada canal:

- Un solo objetivo lógico (que suele ser un fichero de imagen).
- Un identificador asignado por el servidor que habrá de incluirse en cada petición. En JPIP no se define un identificador de sesión independiente, puesto que basta con el identificador de canal para asociar la petición con su sesión.
- Un registro de las capacidades y preferencias del cliente, que puede ajustarse mediante los campos adecuados en la petición.
- Siempre que el servidor ponga en cola las peticiones, podrá proporcionar una cola separada para cada canal JPIP.

Hay una correspondencia uno a uno entre las peticiones y las respuestas de cliente en un canal. Puede haber diferentes canales JPIP en el mismo canal de transporte o en diferentes canales de transporte. Es probable que las peticiones que utilizan diversos canales JPIP lleguen asincrónicamente al servidor cuando se transporten en canales separados. Del mismo modo, cuando se utilicen diversos canales JPIP, las respuestas pueden llegar de forma asíncrona al cliente si se utilizan canales de transporte separados. Aunque el servidor es autónomo al escoger múltiples canales, ha de tener en cuenta el campo petición de velocidad de entrega y las preferencias de ancho de banda máximo y sector de ancho de banda.

B.3 Gestión de modelo caché

Como ya se ha señalado, una de las funciones principales de una sesión consiste en modelar desde el lado del servidor la caché de cliente. A menos que se indique lo contrario, el servidor **puede** suponer que el cliente ha almacenado en caché toda la información enviada como respuesta a sus peticiones en la sesión, es decir, que **no es necesario** retransmitirla. Cabe observar, sin embargo, que el servidor no está obligado a mantener un modelo completo de la caché y puede incluso no tener ninguno: los datos redundantes **pueden** transmitirse al responder las peticiones.

Además de su efecto sobre los datos que se transmiten, es posible que las declaraciones explícitas de manipulación del modelo de caché en las peticiones de cliente actualicen dicho modelo en el servidor. Estas declaraciones se han de procesar antes de establecer cuáles datos se deben retornar al cliente al responder a su petición. Hay dos tipos de declaraciones de manipulación de modelo caché, a saber aditivas y sustractivas.

Las declaraciones aditivas de manipulación del modelo caché permiten aumentar dicho modelo en el servidor al añadir bins de datos o porciones de bins de datos a éste. Gracias a ellas, un cliente puede indicar al servidor qué información ya ha recibido en la sesión anterior o utilizar peticiones previas sin estado. Un servidor **debe** intentar aprovechar toda declaración aditiva de manipulación de modelo caché que figure en una petición de cliente. No obstante, no es obligación de los servidores mantener un modelo caché completo, por lo que pueden ignorar, parcial o totalmente, declaraciones aditivas de manipulación de modelos de caché.

Las declaraciones sustractivas permiten al servidor suprimir bins de datos o porciones de ellos de su modelo caché. Un cliente puede emitir declaraciones sustractivas de manipulación de modelo caché a fin de informar al servidor que no ha almacenado en caché o ha descartado uno de los datos que éste le había enviado antes. De lo contrario, el servidor puede suponer que el cliente ha almacenado todos los datos transmitidos durante la sesión. El servidor **habrá de** suprimir toda información identificada por una declaración sustractiva de modelo caché de cualquier modelo de caché (ya sea completo o no) que esté manteniendo.

Las peticiones JPIP basadas en sesión tienen efectos colaterales que pueden afectar la respuesta a futuras peticiones. Esto también es válido para las peticiones que contengan declaraciones de manipulación de modelo caché, puesto que los efectos de dichas manipulaciones son persistentes. Más aún, los efectos colaterales de una petición que llega a un canal JPIP se reflejan en la respuesta a cualquier petición que puede pertenecer a un canal JPIP diferente que esté asociado con el mismo objetivo lógico. Esto se debe a que hay solamente un modelo caché para cada objetivo lógico en una sesión.

B.4 Interrogación y manipulación de conjuntos modelo

Siempre que un objetivo lógico asociado con una sesión contenga gran cantidad de trenes de código (por ejemplo un objetivo vídeo) o un cliente permanezca conectado durante un largo periodo de tiempo, en las implementaciones prácticas de servidor habrá una tendencia cada vez mayor a utilizar los modelos parciales de caché. Asimismo, cada vez será más probable que los clientes puedan almacenar en caché toda la información recibida del servidor. Para aumentar la eficacia de la comunicación en tales casos, se establece el concepto de un "mset" (conjunto modelo), agrupación de trenes de códigos para los cuales el servidor está modelando el contenido de caché de cliente.

El cliente puede, en cualquier petición, ordenar al servidor que limite su "mset" a determinado conjunto de trenes de código, proporcionando así un mecanismo conveniente para descartar trenes de código completos de su caché sin correr el riesgo de que el servidor genere respuestas incompletas a futuras peticiones de dichos trenes de código.

Las peticiones "mset" también producen respuestas al servidor que indican el conjunto real de trenes de código para los cuales se mantiene información en el modelo caché. Siendo así, los clientes pueden establecer si el servidor ignorará determinadas declaraciones de manipulación del modelo caché que se refieran a una variedad de trenes de código.

De no haber una manipulación o interrogación explícita "mset", el cliente podrá suponer que solamente se incluyen en el "mset" del servidor todos los trenes de código para los cuales se genera información de respuesta a su petición. Debido a que los servidores suelen poder limitar el alcance de una petición de cliente a un número menor de trenes de código que el originalmente especificado, no se puede garantizar que el "mset" del servidor incluirá todos los trenes de código mencionados en una petición, a menos que dicha petición mencione solamente uno de ellos. En C.8.6 se dan más detalles al respecto.

Anexo C

Petición de cliente

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

C.1 Sintaxis de petición

C.1.1 Introducción

En este anexo se describen todos los elementos que pueden formar parte de una petición JPIP. En cada una de las subcláusulas principales se especifica un grupo de campos y sus posibles valores. En general, una petición constará de campos de más de un grupo, aunque algunos grupos son incompatibles. Más aún, dentro de cada grupo puede haber campos de petición incompatibles entre ellos. Algunas peticiones, que en otras circunstancias serían válidas, no son admitidas en ciertas situaciones (por ejemplo sesiones), aunque no haya sido indicado por la sintaxis BNF. Para terminar, aun si la petición es válida, puede ocurrir que el servidor no soporte todos los campos posibles de petición o sus combinaciones.

C.1.2 Estructura de petición

La petición JPIP está compuesta por los siguientes campos:

- Campos de identificación de objetivo.
- Campos de gestión de sesión y canal.
- Campos de petición de ventana seleccionada.
- Campos de metadatos.
- Campos de petición de limitación de datos.
- Campos de petición de control de servidor.
- Campos de petición de gestión de caché.
- Campos de petición de cargado.
- Campos de capacidades y preferencias de cliente.

Se han de enviar los elementos en una petición conforme al protocolo de transporte escogido. Por ejemplo, en HTTP las peticiones se expresan mediante los caracteres enumerados en la sintaxis BNF, los parámetros múltiples se unen mediante un carácter "&", y las peticiones pueden formar parte del campo consulta (query) o de una petición GET, o del cuerpo de una petición POST. En los anexos F, G y H se suministran más detalles al respecto.

NOTA – Es posible reemplazar caracteres reservados URI. Por ejemplo, "request=a:b" en el caso de un URL GET HTTP tendrá como resultado "request=a%3Ab", en donde el carácter reservado URL ":" se ha reemplazado por "%3A".

```

jpip-request-field = target-field
                    / channel-field
                    / view-window-field
                    / metadata-field
                    / data-limit-field
                    / server-control-field
                    / cache-management-field
                    / upload-field
                    / client-cap-pref-field

target-field       = target           ; C.2.2
                   / subtarget       ; C.2.3
                   / tid              ; C.2.4

channel-field      = cid              ; C.3.2
                   / cnew            ; C.3.3
                   / cclose          ; C.3.4
                   / qid             ; C.3.5

```

ISO/CEI 15444-9:2005 (S)

<code>view-window-field</code>	<code>= fsiz</code>	<code>; C.4.2</code>
	<code>/ roff</code>	<code>; C.4.3</code>
	<code>/ rsiz</code>	<code>; C.4.4</code>
	<code>/ comps</code>	<code>; C.4.5</code>
	<code>/ stream</code>	<code>; C.4.6</code>
	<code>/ context</code>	<code>; C.4.7</code>
	<code>/ srates</code>	<code>; C.4.8</code>
	<code>/ roi</code>	<code>; C.4.9</code>
	<code>/ layers</code>	<code>; C.4.10</code>
<code>metadata-field</code>	<code>= metareq</code>	<code>; C.5.2</code>
<code>data-limit-field</code>	<code>= len</code>	<code>; C.6.1</code>
	<code>/ quality</code>	<code>; C.6.2</code>
<code>server-control-field</code>	<code>= align</code>	<code>; C.7.1</code>
	<code>/ wait</code>	<code>; C.7.2</code>
	<code>/ type</code>	<code>; C.7.3</code>
	<code>/ drate</code>	<code>; C.7.4</code>
<code>cache-management-field</code>	<code>= model</code>	<code>; C.8.1</code>
	<code>/ tpmodel</code>	<code>; C.8.3</code>
	<code>/ need</code>	<code>; C.8.4</code>
	<code>/ tpneed</code>	<code>; C.8.5</code>
	<code>/ mset</code>	<code>; C.8.6</code>
<code>upload-field</code>	<code>= upload</code>	<code>; C.9.1</code>
<code>client-cap-pref-field</code>	<code>= cap</code>	<code>; C.10.1</code>
	<code>/ pref</code>	<code>; C.10.2</code>
	<code>/ csf</code>	<code>; C.10.3</code>

C.1.3 Restricciones a la combinación de campos de petición

En una misma petición sólo podrá aparecer una vez cada tipo de campo de petición JPIP.

En general, se podrán combinar peticiones de datos de imagen (peticiones de ventana seleccionada) con peticiones de metadatos adicionales. No obstante, cómo se combinen estos campos está sujeto a restricciones.

El campo de petición de cargado no se combinará con los campos `metadata-field`, `data-limit-field` o `server-control-field`.

C.2 Campos de identificación de objetivo

C.2.1 Introducción a los objetivos lógicos

Cada petición JPIP se dirige a determinada representación de un recurso designado original específico o a una porción específica de él. Este recurso puede ser un fichero u objeto almacenado físicamente o algo que el servidor cree virtualmente tras la petición.

Se denomina objetivo lógico a la representación específica, ya sea ésta la forma codificada original o una transcodificada, o se trate bien de un intervalo específico de bits o bien del recurso entero. El objetivo lógico se especifica mediante tres campos de petición: Target ID, Target y Sub-target.

El campo de petición Target especifica el recurso designado original al cual se dirige la petición. Se especifica mediante un `PATH`, que puede tratarse de una cadena simple o de un URI. Si no se especifica el campo Target y la petición se efectúa a través de HTTP, la petición JPIP se dirigirá al recurso especificado a través de la componente de trayecto del URL de petición JPIP. Este recurso designado original puede tratarse de un fichero real u otro objeto físico almacenado en el servidor o puede también ser algo creado por el servidor como respuesta a una petición JPIP.

El campo de petición Sub-target especifica el intervalo preciso de bytes del recurso designado original (especificado a su vez mediante el campo de petición Target) al que se dirige la petición. Si no se especifica, la petición va dirigida a todo el intervalo de bytes del recurso original.

El campo de petición Target ID se puede utilizar para especificar aún más determinada codificación del recurso en situaciones en las que el cliente y servidor hayan ya intercambiado datos de él. Por ejemplo, es posible que el servidor haya suministrado previamente una versión transcodificada del fichero al cliente basándose en información proporcionada y en las condiciones de una petición anterior. Si el cliente ha conservado los datos transmitidos anteriormente en su caché, querrá seguir recibiendo datos con la misma transcodificación para poder utilizar los ya

almacenados. El Target ID es una cadena de identificación definida por el servidor a la que éste ha asociado anteriormente una representación específica de dicho recurso designado original o un intervalo de bytes de algún recurso designado original específico.

Si un cliente especifica tanto el recurso designado original (ya sea a través del campo petición Target o del componente de trayecto del URL de petición JPIP) y el Target ID, el servidor habrá de verificar si puede responder a la petición de la misma manera como si él mismo hubiese asignado dicho Target ID al recurso. Si el servidor no puede hacerlo, utilizará el encabezamiento respuesta JPIP tid para informar al cliente el nuevo Target ID, con lo cual éste sabrá que tiene que descartar todo dato previamente almacenado en la caché.

Si se ha de ofrecer a un objetivo lógico mensajes de tren JPP o JPT, los bins de datos asociados serán coherentes a través de todas las respuestas que se emitan en la misma sección. Cuando el servidor, o un servidor relacionado con él, emita también un Target ID, los bins de datos permanecerán coherentes a través de todas las respuestas emitidas con el mismo Target ID, sin importar si han sido emitidos en la misma sesión o no.

Si esta petición forma parte de una sesión y si el servidor ha atribuido un ID de canal, el cliente podrá especificar este ID a través del campo de petición Channel ID en lugar de especificar el campo de petición Target, Sub-target y Target ID. Cuando se especifique el objetivo lógico mediante una combinación de campos Target, Sub-target y Target ID y a través del campo de petición Channel ID, el servidor responderá con un mensaje de error.

En los siguientes ejemplos se muestra la especificación de objetivo lógico:

EJEMPLO 1: Para el URL de petición JPIP

"http://one.jpeg.org/imageserver.cgi?target= http%3A%2F%2Fone.jpeg.org%2Fimages%2Fpicture.jp2&fsiz=200,200" el objetivo lógico es toda la gama de bytes contenida dentro del URI "http://one.jpeg.org/images/picture.jp2," relativo al directorio raíz de documentos del servidor.

EJEMPLO 2: Para el URL de petición JPIP

"http://one.jpeg.org/imageserver.cgi? target= http%3A%2F%2Fone.jpeg.org%2Fimages%2Fpicture.jp2&tid=4384-5849-af4d-3dca&fsiz=200,200" el objetivo lógico es la gama completa de bytes contenida dentro del URI "http://one.jpeg.org/images/picture.jp2," relativo al directorio raíz de documentos de servidor, con una representación especificada por el Target ID definido por el servidor ID 4384-5849-af4d-3dca.

EJEMPLO 3: Para el URL de petición JPIP

"http://one.jpeg.org/imageserver.cgi?target= http%3A%2F%2Fone.jpeg.org%2Fimages%2Fpicture.jp2&subtarget=1038-13458&fsiz=200,200" el objetivo lógico es la gama de bytes, empezando por el byte 1038, y siguiendo hasta 13458, contenido dentro del URI "http://one.jpeg.org/images/picture.jp2," relativo al directorio raíz de documentos de servidor.

EJEMPLO 4: Para el URL de petición JPIP de "http://one.jpeg.org/imageserver.cgi?cid=1234-5849-af4d-3dca&fsiz=200,200" el objetivo lógico es el recurso al que el servidor ha asociado con el canal cuyo ID es 1234-5849-af4d-3dca.

EJEMPLO 5: Para el URL de petición JPIP de "http://one.jpeg.org/images/picture.jp2?fsiz=200,200" el objetivo lógico es la gama completa de bytes contenida dentro del fichero "images/picture.jp2," relativo al directorio raíz de documentos de servidor.

EJEMPLO 6: Para el URL de petición JPIP de "http://one.jpeg.org/images/picture.jp2?subtarget=1038-13458&fsiz=200,200" el objetivo lógico es la gama de bytes, empezando desde el byte 1038, y hasta el 13458 inclusive, contenido dentro del fichero "images/picture.jp2," relativo al directorio raíz de documentos del servidor.

C.2.2 Objetivo (*target*)

```
target = "target" "=" PATH
```

Este campo se utiliza para especificar el recurso designado original (que suele ser el nombre de un fichero en el servidor). Cuando no hay campo de petición Target se determina el recurso designado original mediante otro método.

C.2.3 Subjetivo (*subtarget*)

```
subtarget = "subtarget" "=" byte-range
byte-range = UINT-RANGE
```

Este campo se puede utilizar para calificar el recurso designado original mediante la especificación de una gama de bytes. El objetivo lógico se ha de interpretar como la gama indicada de bytes del recurso designado original.

Los límites superior e inferior de la gama de bytes suministrada se incluyen y el número 0 se refiere al primer byte del fichero objetivo.

C.2.4 ID de objetivo (*tid*, o *target-id*)

```
tid = "tid" "=" target-id  
target-id = TOKEN
```

Este campo se puede utilizar para proporcionar una cadena *target-id*, que el servidor ha generado previamente para identificar completamente el objetivo lógico al que se está accediendo, incluida toda transcodificación discrecional que haya efectuado el servidor. Si bien el nombre del objetivo lógico no tiene que ser único y no es necesario que corresponda a una única codificación de su contenido, la cadena *target-id*, junto con el nombre y la gama de bytes del recurso original han de identificar completamente la imagen y su codificación.

Cuando *target-id* sea "0", el objetivo lógico se especifica mediante la utilización del Target, Sub-target y del componente de trayecto URL JPIP, y el cliente estará solicitando explícitamente que el servidor informe el *target-id* atribuido, de haberlo. El servidor incluirá un encabezamiento Target ID en su respuesta a toda petición de cliente cuyo *target-sea* sea "0".

La longitud de *target-id* será a lo sumo 255.

C.3 Campos para el trabajo con sesiones y canales

C.3.1 Introducción

Una petición deberá ser sin estado a menos que ocurra una de las condiciones siguientes o ambas:

- La petición incluye un campo válido Channel ID;
- La petición incluye un campo New Channel (véase más adelante) y la respuesta del servidor incluye un encabezamiento de respuesta New Channel cuyo *channel-id* viene de ser emitido.

En B.2 se presentan más detalles sobre las sesiones y canales.

C.3.2 ID de canal (*cid*, o *channel ID*)

```
cid = "cid" "=" channel-id  
channel-id = TOKEN
```

- Este campo se utiliza para asociar la petición con un canal JPIP determinado y, por ende, la sesión a la cual éste pertenece.

C.3.3 Nuevo canal (*cnew*)

```
cnew = "cnew" "=" l#transport-name  
transport-name = TOKEN
```

Este campo se utiliza para solicitar un nuevo canal JPIP. De no haber campo de petición Channel ID, la petición se hace para una nueva sesión. De lo contrario, se pide un nuevo canal en la misma sesión identificado por el campo de petición Channel ID.

La cadena de valor identifica los nombres de uno o varios protocolos de transporte que el cliente esté dispuesto a aceptar. En esta Recomendación | Norma Internacional se definen solamente los nombres de transporte "http" y "http-tcp", aunque cabe esperar que se puedan definir otros tipos de transporte como "udp". En el anexo F se presentan más detalles acerca de la utilización de JPIP con el transporte "http", mientras que en el anexo G se discute dicha utilización con el transporte "http-tcp".

Si el servidor desea abrir un nuevo canal, utilizando uno de los protocolos de transporte indicados, tendrá que retornar el testigo identificador de nuevo canal a través del encabezamiento de respuesta New Channel (véase D.2.3). En este caso, la petición en curso es la primera en el nuevo canal.

Un cliente puede abrir un canal a un nuevo objetivo lógico dentro de la misma sesión, para lo cual habrá de identificar tanto un Channel ID existente como un objetivo lógico. Al abrir un nuevo canal al mismo objetivo lógico que se asocia con un canal existente no es necesario especificar explícitamente el objetivo lógico.

Si el servidor no desea abrir un nuevo canal, no retornará el encabezamiento de respuesta New Channel sino que solicitará que se lo atienda como si el campo de petición New Channel no hubiese sido incluido. Es decir, una petición que especifique un Channel ID existente se tratará como una petición en dicho canal, mientras que una que no incluya dicho campo se tratará como una petición sin estado. Siempre que la petición New Channel identifique un objetivo

lógico diferente al que se ha asociado con el Channel ID existente proporcionado, el servidor no podrá responder a ésta sin haber bien emitido un nuevo Channel ID o retornado un código de error.

EJEMPLO 1: "target=nice.jp2&cnew=http" es la petición del primer canal de una nueva sesión con la imagen "nice.jp2" utilizando el transporte "http". Si el servidor no atribuye canal, la petición se tratará como sin estado.

EJEMPLO 2: "cid=013ac8&cnew=http-tcp" es la petición de un nuevo canal en la misma sesión que se ha asociado con el Channel ID 013ac8. El nuevo canal habrá de utilizar el transporte "http-tcp" y referirse al mismo objetivo lógico que el Channel ID 013ac8. Estos canales comparten un sólo modelo caché. Si el servidor no atribuye canal, la respuesta se tratará como si se hubiese omitido el campo de petición New Channel.

EJEMPLO 3: "target=nice.jp2&cid=013ac8&cnew=http" es la petición de un nuevo canal en la misma sesión que está asociada con el Channel ID "013ac8." El nuevo canal habrá de utilizar el transporte "http". El objetivo lógico asociado con el nuevo canal es diferente de aquel asociado con el Channel ID "013ac8" y se ha de utilizar para el nuevo canal un modelo caché independiente. Los modelos cache de ambos objetivos se asocian con su sesión común.

C.3.4 Cerrar canal (*cclose* o *Channel Close*)

```
cclose = "cclose" "=" ("*" / 1#channel-id)
```

Este campo se utiliza para cerrar uno o varios canales abiertos hacia una sesión. Si el campo valor contiene uno o varios testigos Channel ID, éstos deberán pertenecer a la misma sesión. En este caso, no es necesario el campo de petición Channel ID aunque si lo hay deberá hacer también referencia a un canal que forme parte de la misma sesión.

Si el campo valor es "*", todos los canales asociados con la sesión se cerrarán, identificándola así mediante la inclusión de un campo petición Channel ID.

El servidor completará su respuesta en todo canal especificado en la petición Channel Close antes de cerrar realmente dicho canal.

C.3.5 ID de petición (*qid* o *Request ID*)

```
qid = "qid" "=" UINT
```

Este campo se utiliza para especificar un valor Request ID. Cada canal posee su propia cola de petición, con su propio contador de Request ID. Las peticiones que se reciban dentro de cualquier canal determinado (indicado por el valor Channel ID) se tramitarán en el orden de sus valores Request ID, siempre que se utilice dicho campo. El servidor podrá procesar peticiones que no contengan un campo Request ID a medida que vayan llegando. No obstante, no procesará una petición que llegue con un valor Request ID n hasta tanto no haya procesado todas las peticiones cuyos valores Request ID sean menores que n , y estén asociadas al mismo canal, a menos que $n=0$. El cliente no emitirá una petición que especifique el mismo valor Request ID de cualquier otra petición asociada con el mismo canal, y no emitirá Request ID menores que cualquier otro previamente emitido en el mismo canal.

C.4 Campos de petición de ventana seleccionada

C.4.1 Correspondencia de peticiones de ventana seleccionada con resoluciones y regiones de imagen de tren de código

JPIP tiene como fin proporcionar porciones de imagen JPEG 2000 y metadatos asociados como respuesta a peticiones provenientes de un cliente. Esto se hace mediante una secuencia de peticiones y respuestas. Para una porción de imagen, los datos solicitados pueden ser menores que la imagen total en términos de tamaño de cuadro de imagen, región, calidad y/o componentes.

En el caso más simple, la porción de imagen en cuestión se define directamente con respecto a la rejilla de referencia de alta resolución de los trenes de código JPEG 2000 identificados en la petición, y no a la rejilla de muestra de cualquier imagen determinada de componente. No obstante, los clientes pueden solicitar objetos de imagen de más alto nivel (por ejemplo capas de composición JPX o pistas de vídeo MJ2) a través del campo de petición Codestream Context (véase C.4.7). En este caso, tal vez sea necesario someter a la porción de imagen solicitada a una transformación de coordenadas con el fin de determinar la porción de cada tren de código asociado que se solicita. Dichas transformaciones se describen en C.4.7, y deben entenderse en términos de la siguiente descripción de regiones de imagen de tren de código.

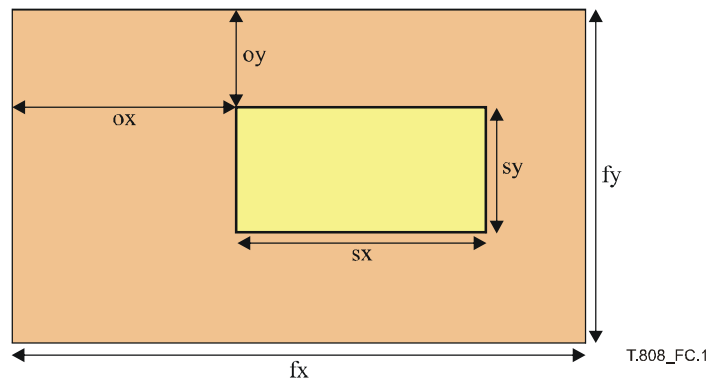


Figura C.1 – Región deseada dentro de una imagen

Las regiones de imagen de tren de código se describen mediante 3 parámetros bidimensionales como se muestra en la figura C.1. Los parámetros de tamaño (s_x y s_y) y los de desplazamiento (o_x y o_y) especifican el ancho y el alto de la región de imagen de tren de código deseada y su esquina superior izquierda, con respecto a una imagen total que tiene el tamaño de trama dado (f_x y f_y).

EJEMPLO 1: Un cliente que desee llenar completamente una pantalla de 640×480 con toda la imagen puede efectuar la siguiente petición "`fsize=640,480&rsiz=640,480&roff=0,0`". Cabe observar que esto se puede hacer independientemente del tamaño original de la imagen (y, por ende, sin necesidad de conocerlo).

Cuando ninguna de las resoluciones de imágenes disponible en los trenes de código JPEG 2000 corresponda exactamente al tamaño de cuadro solicitado, puede ocurrir que la imagen devuelta sea mayor o menor que dicho tamaño e incluso que sea diferente en su formato de imagen. El servidor habrá de establecer una resolución de imagen de tren de código adecuada, que viene dada por los parámetros de tamaño f_x' y f_y' , y una **región del tren de código adecuada** correspondiente a los parámetros s_x' , s_y' , o_x' y o_y' , como se muestra en la figura C.2. Aunque es posible que el cliente especifique el sentido de la aproximación, como parte del campo de petición Frame Size (tamaño de cuadro), tendrá que estar listo para procesar datos devueltos que no correspondan exactamente con los parámetros solicitados.

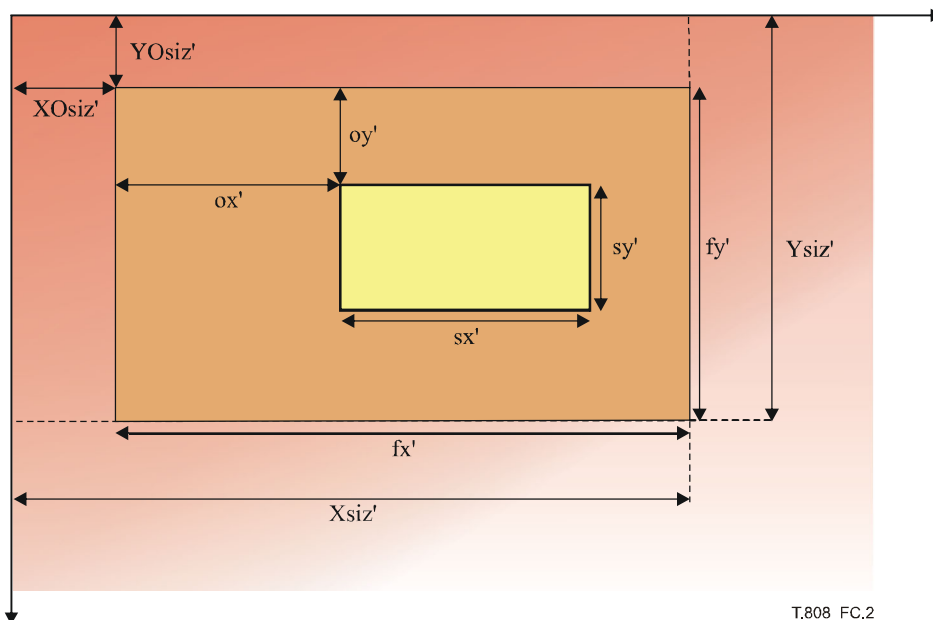


Figura C.2 – Región deseada con respecto a la rejilla de referencia submuestreada

Como se muestra en la figura C.2, el tamaño de la resolución de imagen de tren de código adecuada viene dado por $f_x' = Xsiz' - XOsiz'$ y $f_y' = Ysiz' - YOsiz'$, donde $XOsiz'$, $YOsiz'$, $Xsiz'$, e $Ysiz'$ se calculan utilizando la ecuación C-1.

$$XO_{siz}' = \left\lceil \frac{XO_{siz}}{2^r} \right\rceil; \quad YO_{siz}' = \left\lceil \frac{YO_{siz}}{2^r} \right\rceil; \quad Xs_{siz}' = \left\lceil \frac{Xs_{siz}}{2^r} \right\rceil; \quad Ys_{siz}' = \left\lceil \frac{Ys_{siz}}{2^r} \right\rceil \quad (C-1)$$

donde:

r es calculado por el servidor con el fin de obtener la mejor aproximación al tamaño de imagen solicitado (f_x y f_y), sujeto a las preferencias de aproximación suministradas a través del campo de petición Frame Size.

XO_{siz} , YO_{siz} , Xs_{siz} e Ys_{siz} se obtienen aquí a partir del segmento marcador SIZ del tren de código pertinente. Conviene interpretar r como la cantidad de niveles DWT más altos descartados y, por tanto, r tiene que ser un entero mayor que 0. Sin embargo, el valor de r no está limitado por la cantidad de niveles DWT que han sido utilizados para comprimir cualquier componente losa en el tren de código.

Una vez encontrado el tamaño de cuadro adecuado, f_x' y f_y' , se determinan mediante la ecuación C-2 el tamaño de imagen s_x' y s_y' , y el desplazamiento ox' y oy' , asociados con la región de imagen de tren de código (C.2).

$$ox' = \left\lfloor ox \cdot \frac{f_x'}{f_x} \right\rfloor; \quad oy' = \left\lfloor oy \cdot \frac{f_y'}{f_y} \right\rfloor; \quad sx' = \left\lfloor (sx + ox) \cdot \frac{f_x'}{f_x} \right\rfloor - ox'; \quad sy' = \left\lfloor (sy + oy) \cdot \frac{f_y'}{f_y} \right\rfloor - oy' \quad (C-2)$$

EJEMPLO 2: Supóngase que el tamaño de cuadro solicitado es 128×128 , y que la imagen en la rejilla de referencia de alta resolución de tren de código se describe mediante $XO_{siz}=127$, $Xs_{siz}=648$, $YO_{siz}=0$ e $Ys_{siz}=504$. Asimismo, supóngase que existen 3 niveles de transformada en ondícula discreta para todas las componentes en el tren de código. Los tamaños de imagen de tren de código expandibles son entonces:

$$\begin{aligned} 521 \times 504 & \left(\left\lceil \frac{648}{1} \right\rceil - \left\lceil \frac{127}{1} \right\rceil \text{ por } \left\lceil \frac{504}{1} \right\rceil - 0 \right) \\ 260 \times 252 & \left(\left\lceil \frac{648}{2} \right\rceil - \left\lceil \frac{127}{2} \right\rceil \text{ por } \left\lceil \frac{504}{2} \right\rceil - 0 \right) \\ 130 \times 126 & \left(\left\lceil \frac{648}{4} \right\rceil - \left\lceil \frac{127}{4} \right\rceil \text{ por } \left\lceil \frac{504}{4} \right\rceil - 0 \right) \\ 65 \times 63 & \left(\left\lceil \frac{648}{8} \right\rceil - \left\lceil \frac{127}{8} \right\rceil \text{ por } \left\lceil \frac{504}{8} \right\rceil - 0 \right) \end{aligned}$$

Es decir, si se le solicita un tamaño mayor de cuadro (*round-direction* es *round-up*) el tamaño de cuadro devuelto será 260×252 , de lo contrario (*round-direction* es *round-down*), se utilizará un tamaño de cuadro de 65×63 . Obsérvese que, como ocurre en este ejemplo, los tamaños de cuadro de imagen de tren de código suelen no ser potencias exactas de 2.

El submuestreo de una componente de imagen, como se especifica mediante XR_{siz} e YR_{siz} , no influye en la interpretación de la región de imagen o resolución de imagen solicitadas dentro de un tren de código pedido.

EJEMPLO 3: Se puede efectuar una petición de una región de 256×256 del extremo superior izquierdo de una imagen de 512×512 mediante la instrucción: `fsiz=512,512&rsiz=256,256`.

Supóngase que el tren de código contiene una imagen dividida en submuestras que resulta en las componentes 1 y 2 más no en la componente 0. En particular, sea $Xs_{siz}=1024$, $Ys_{siz}=1024$, $XO_{siz}=0$, $YO_{siz}=0$, y $XR_{siz}^0=1$, $YR_{siz}^0=1$, $XR_{siz}^1=2$, $YR_{siz}^1=2$, $XR_{siz}^2=2$, e $YR_{siz}^2=2$. Aunque en este caso el servidor excluirá el más alto nivel de resolución de las tres componentes y retornará losas o recintos suficientes para proporcionar 256×256 muestras de la componente 0, para las componentes 1 y 2 habrá solamente 128×128 muestras. El cliente dispondrá entonces de datos para presentar en pantalla la esquina superior izquierda de toda la imagen que aún está dividida en muestras. Cuando el cliente desee obtener componentes cromáticos no muestreados deberá emitir una petición adicional del tipo:

`fsiz=1024,1024&rsiz=512,512&comps=1,2`

En tal caso, el servidor retornará datos suficientes para suministrar 256×256 muestras de las componentes 1 y 2 que puedan combinarse con los datos de la componente 0 ya recibidos para obtener una imagen no muestreada aunque de la mitad del tamaño de la imagen.

Si las tres componentes ya han sido submuestreadas, el servidor proporcionará solamente 128×128 muestras de las tres componentes para la petición original ($fsiz=512, 512$ y $rsiz=256, 256$) puesto que tanto la resolución como las regiones de la imagen se evalúan con respecto a la rejilla de referencia de cada tren de código solicitado.

C.4.2 Tamaño de cuadro (*fsiz* o, *Frame size*)

```
fsiz = "fsiz" "=" fx "," fy ["", "round-direction"]
fx = UINT
fy = UINT
round-direction = "round-up" / "round-down" / "closest"
```

Este campo se utiliza para identificar la resolución correspondiente a la ventana seleccionada solicitada. Los valores *fx* y *fy* especifican las dimensiones de la resolución de imagen deseadas. El valor *round-direction* especifica cómo se puede escoger un tren de código disponible para cada tren de código solicitado cuando la resolución de imagen solicitada no esté disponible en cada tren de código. Se hace corresponder el tamaño de cuadro solicitado con una resolución de imagen de tren de código mediante el procedimiento descrito en C.4.1, añadiéndole tal vez las transformaciones de coordenadas pedidas a través de un campo de petición Codestream Context (véase C.4.7). Es posible que un cliente que desee controlar la cantidad exacta de muestra recibida para determinada componente de imagen deba incrementar el tamaño de imagen solicitado, tal como se explica en C.4.1. En el cuadro C.1 se describen las opciones *round-direction* definidas en esta Recomendación | Norma Internacional.

Cuadro C.1 – Opciones de aproximación

Round-direction	Significado
"round-up"	Para cada tren de código solicitado, se ha de escoger la menor resolución de imagen de tren de código cuyo ancho y alto sean mayores o iguales que el tamaño especificado. De no haberla, se utilizará la mayor resolución de imagen de tren de código disponible.
"round-down"	Para cada tren de código solicitado, se escogerá la mayor resolución de imagen de tren de código cuyos ancho y alto sean menores o iguales que el tamaño especificado. Cuando no se especifique el parámetro <i>round-direction</i> éste es el valor por defecto.
"closest"	Para cada tren de código solicitado, se escogerá la resolución de imagen de tren de código que sea más próxima (<i>closest</i>) a la superficie especificada (donde superficie = $fx \times fy$). Cuando haya dos resoluciones de imagen de tren de código que tengan superficies equidistantes de $fx \times fy$, se escogerá la mayor de las dos.

Si se omite el campo de petición tamaño de cuadro de una petición de ventana seleccionada y si no se especifica *metadata-only* en un campo de petición de metadatos (véase C.5.1), la ventana seleccionada solicitada no incluye datos de imagen comprimida ni encabezamientos específicos de losa, aunque sí toda la otra información de encabezamiento (tren de código y formato de fichero) que se habría retornado de haber incluido el cliente el campo de petición de tamaño de cuadro. En C.5.1 se presenta más información acerca del formato de ficheros metadatos que se solicita implícitamente junto con la petición de ventana seleccionada.

C.4.3 Desplazamiento (*roff*, u *Offset*)

```
roff = "roff" "=" ox "," oy
ox = UINT
oy = UINT
```

Este campo se utiliza para identificar la esquina superior izquierda (desplazamiento) de la región espacial correspondiente a la ventana seleccionada solicitada; de no haberlo, el valor por defecto del desplazamiento es 0. El desplazamiento real de una región de imagen de tren de código a partir de la esquina superior izquierda de la imagen y la resolución real de imagen de tren de código escogida por el servidor se obtienen siguiendo el procedimiento descrito en C.4.1, añadiendo posiblemente las transformaciones de coordenadas que se soliciten a través del campo de petición contexto de tren de código (Codestream Context) (véase C.4.7).

El campo desplazamiento es válido sólo si se utiliza junto con el campo de petición tamaño de cuadro.

Cuando la región de imagen de tren de código especificada mediante el tamaño y/o desplazamiento de región resulta vacía (sin superficie), la respuesta del servidor no incluirá ninguna información de imagen comprimida para dicho tren de código. En particular, las respuestas del tipo tren JPP o JPT no incluirán mensajes que hagan referencia a los bins de datos de recinto, losa o encabezamiento de losa de dicho tren de código. Es facultad del servidor devolver mensajes de

encabezamiento de datos o bin de metadatos que habrían sido devueltos como respuesta a una petición en la que se haya omitido el campo petición tamaño de trama.

C.4.4 Tamaño de región (*rsiz* o *Region Size*)

```
rsiz = "rsiz" "=" sx "," sy
sx = UINT
sy = UINT
```

Este campo se utiliza para identificar el tamaño horizontal y vertical de la región espacial correspondiente a la ventana seleccionada solicitada. De no haberlo, la región va hasta la esquina inferior derecha de la imagen. Se calculan las dimensiones reales de una región de imagen de tren de código, a la resolución real de imagen de tren de código escogida por el servidor, siguiendo el procedimiento descrito en C.4.1, con la posible adición de transformaciones de coordenadas solicitadas a través del campo de petición contexto de tren de código (véase C.4.7). No es necesario que una región de imagen de tren de código solicitada esté incluida completamente dentro del tren de código, en cuyo caso el servidor tomará simplemente la intercepción entre la región de imagen de tren de código disponible y la región solicitada.

El campo de petición tamaño de región es válido solamente cuando se usa junto con el campo de petición tamaño de cuadro.

La región de imagen de tren de código puede ser vacía, por ejemplo cuando *sx* o *sy* sean cero. De serlo, la respuesta del servidor no incluirá entonces ningún dato de imagen comprimida para dicho tren de código. En particular, las respuestas a los trenes de tipo JPP o JPT no contendrán mensajes que hagan referencia a los bins de datos de recinto, losa o encabezamiento de losa de dicho tren de código. Es facultad del servidor retornar mensajes de encabezamiento principal o bins de metadatos que hubiera retornado en caso de respuesta a una petición en la que se hubiera omitido el campo petición tamaño de cuadro.

C.4.5 Componentes (*comps*)

```
comps = "comps" "=" 1#UINT-RANGE
```

Este campo se utiliza para identificar las componentes de imagen que se ha de incluir en la ventana seleccionada solicitada. De no haberlo, se entiende que la petición incluye todas las componentes disponibles de imagen de todos los trenes de código identificados a través del campo de petición tren de código y todas las componentes pertinentes de todos los trenes de códigos solicitados mediante el campo de petición contexto de tren de código (véase C.4.7). Dichas componentes "pertinentes" son aquellas que participan en la reproducción de las entidades de imagen (por ejemplo, capas de composición JPX o pistas de vídeo MJ2) que se especifica mediante el campo de petición tren de código.

Los valores en dicho campo de petición representan los índices de las componentes de imagen de interés, que empiezan desde 0 y que se interpretan utilizando la sintaxis de tren de código JPEG 2000, tal como se describe en la Rec. UIT-T T.800 | ISO/CEI 15444-1, aunque cabe observar que son las componentes que se obtienen al decodificar y aplicar la transformada en onícula discreta inversa a los datos comprimidos, antes de aplicar la transformada de componentes inversa RCT o ICT. Para los trenes de código conformes a la Rec. UIT-T T.801 | ISO/CEI 15444-2, las componentes que se identifican aquí son aquellas correspondientes a "componentes espaciales", es decir las que se obtienen al decodificar y aplicar la transformada en onícula discreta inversa a los datos comprimidos, antes de aplicar cualquier otra transformada multicomponentes inversa, transformada de componentes de dependencia o transformada en onícula discreta multicomponentes.

Se descartarán todas las componentes no existentes en cualquier tren de código solicitado.

C.4.6 Tren de códigos (*stream* o *Codestream*)

```
stream = "stream" "=" 1#sampled-range
sampled-range = UINT-RANGE [":" sampling-factor]
sampling-factor = UINT
```

Este campo se utiliza para identificar el tren o los trenes de código que pertenecen a la ventana seleccionada solicitada. Cuando se omita este campo y no se puedan determinar por otros medios los trenes de código, su valor por defecto es el tren de código único cuyo identificador es 0. Obsérvese que el campo petición contexto de tren de código (véase C.4.7) proporciona un método adicional para solicitar trenes de código.

En el caso de objetivos de la familia JPEG 2000, los índices de tren de código son aquéllos incorporados en la correspondiente casilla marcador de posición que aparece dentro del bin de metadatos adecuado, como se describe en A.3.6. Para los formatos de fichero que lleven implícitos los identificadores de tren de código, dichos identificadores habrán de corresponder con los índices utilizados aquí.

Siempre que se identifique una gama de trenes de código, la no existencia de un límite superior indica que ésta se extiende a todos los trenes de código que tengan identificadores mayores. De lo contrario, el límite superior proporciona el identificador absoluto del último tren de códigos en la gama.

Independientemente de si indica o no un límite superior, es posible calificar una gama de trenes de código mediante un *sampling-factor* adicional que, de haberlo, será estrictamente un entero positivo, F . En tal caso, la gama incluye todos los identificadores de tren de código $L+Fk$ que caen en dicho rango no calificado, en donde L es el identificador del primer tren de código en dicha gama. El índice del cliente del tren de código de interés es k y k es una UINT.

C.4.7 Contexto de tren de código (*context* o *Codestream Context*)

```
context = "context" "=" 1#context-range
context-range = jpx1-context-range / mj2t-context / reserved-context
jpx1-context-range = "jpx1" "<" jpx-layers ">" [ "[" jpx1-geometry "]" ]
jpx-layers = sampled-range
jpx1-geometry = "s" jpx-iset "i" jpx-inum
jpx-iset = UINT
jpx-inum = UINT
mj2t-context = "mj2t" "<" mj2-track ">" [ "[" mj2t-geometry "]" ]
mj2-track = NONZERO ["+" "now" ]
mj2t-geometry = "track" / "movie"
reserved-context = 1*( TOKEN / "<" / ">" / "[" / "]" / "-" / ":" / "+" )
```

Este campo se puede utilizar para solicitar indirectamente trenes de código a través de entidades de imagen de "nivel superior". En esta Recomendación | Norma Internacional se definen los contextos correspondientes a las capas de composición JPX (capas que pueden involucrar uno o varios trenes de código) y a las pistas de vídeo MJ2; sin embargo, el mecanismo se diseña con el fin de poder ampliarlo ulteriormente.

Cuando hay un campo petición de contexto de tren de código, la ventana seleccionada solicitada incluye cada uno de los trenes de código que se asocia con él o los contextos solicitados, además de todos los trenes de código que hayan sido solicitados a través del campo de petición de tren de código.

El cuerpo de un campo de petición de contexto de tren de código consta de uno o varios valores *context-range*, cada uno de los cuales corresponde a un conjunto de trenes de código que ha sido determinado por el servidor. Del mismo modo, es posible que un *context-range* identifique también transformaciones de recorrespondencia de coordenadas que se deban aplicar a los parámetros tamaño de cuadro, tamaño de región y desplazamiento con el fin de establecer la resolución de imagen y la región de imagen del tren de código para cada uno de los trenes de código asociados con dicho *context-range*. Si el servidor puede procesar un *context-range*, tendrá que identificar los trenes de código asociados con él mediante un encabezamiento de respuesta de contexto de tren de código.

En esta Recomendación | Norma Internacional se definen dos tipos específicos de *context-range*, destinados a los formatos de fichero JPX y MJ2. El primero de estos tipos *context-range*, *jpx1-context-range*, se utiliza para identificar una o varias capas de composición JPX, cuyos índices asociados con un *jpx1-context-range* se suministran como un *sampled-range*, conforme a la misma semántica de las gamas de trenes de código muestreadas en el campo de petición de tren de código. Cuando el servidor procese un *jpx1-context-range*, los trenes de código que pertenezcan a la o las capas de composición correspondientes se identificarán mediante un encabezamiento de respuesta contexto de tren de código.

Un *jpx1-context-range* puede identificar la transformación opcional de recorrespondencia de coordenadas que se ha de utilizar al establecer la resolución de imagen y la región de imagen del tren de código para cada uno de sus trenes de código. Esta transformación de recorrespondencia viene dada por dos enteros no negativos, *jpx-iset* y *jpx-inum*, que identifican una instrucción específica de composición dentro de una casilla de composición (comp) JPX, que se encuentra dentro del alcance del objetivo lógico. Dicha instrucción específica está ubicada en la casilla de conjunto de instrucción (iset) cuya posición ordinal (empezando desde 0) dentro de la casilla de composición viene dada por el valor *jpx-iset*. El valor *jpx-inum* corresponde a la posición ordinal (empezando desde 0) de la instrucción dentro de dicha casilla de conjunto de instrucciones. La interpretación de dichos índices es independiente de las repeticiones en el conteo que puedan aparecer dentro de una casilla de composición JPX.

Cuando el servidor tramita valores `jpx-isset` y `jpx-inum`, se han de hacer corresponder en primer lugar los parámetros de tamaño de cuadro y región solicitados f_x , f_y , s_x , s_y , o_x y o_y , a los parámetros modificados de tamaño de cuadro y región f_x'' , f_y'' , s_x'' , s_y'' , o_x'' y o_y'' utilizando la ecuación C-3. Se habrá de calcular cada uno de estos parámetros de región modificada independientemente para cada tren de código solicitado y hay que utilizarlos en lugar de f_x , f_y , s_x , s_y , o_x y o_y cuando se establezca la resolución de imagen y la región de imagen de tren de código conforme al procedimiento que se describe en C.4.1.

$$\begin{aligned}
 f_x'' &= \left[f_x \cdot \frac{X_{R_{reg}}}{X_{S_{reg}}} \cdot \frac{W_{t_{inst}}}{W_{S_{inst}}} \cdot \frac{W_{cod}}{W_{comp}} \right]; & f_y'' &= \left[f_y \cdot \frac{Y_{R_{reg}}}{Y_{S_{reg}}} \cdot \frac{H_{t_{inst}}}{H_{S_{inst}}} \cdot \frac{H_{cod}}{H_{comp}} \right] \\
 s_x'' &= \min \{ (o_x + s_x), x_{lim} \} - \max \{ o_x, x_{min} \} \\
 s_y'' &= \min \{ (o_y + s_y), y_{lim} \} - \max \{ o_y, y_{min} \} \\
 o_x'' &= \max \{ o_x, x_{min} \} - \left[\left(X_{O_{inst}} - \left(X_{C_{inst}} - \frac{X_{O_{reg}}}{X_{S_{reg}}} \right) \cdot \frac{W_{t_{inst}}}{W_{S_{inst}}} \right) \cdot \frac{f_x}{W_{comp}} \right] \\
 o_y'' &= \max \{ o_y, y_{min} \} - \left[\left(Y_{O_{inst}} - \left(Y_{C_{inst}} - \frac{Y_{O_{reg}}}{Y_{S_{reg}}} \right) \cdot \frac{H_{t_{inst}}}{H_{S_{inst}}} \right) \cdot \frac{f_y}{H_{comp}} \right] \\
 x_{min} &= \left[X_{O_{inst}} \cdot \frac{f_x}{W_{comp}} \right]; & y_{min} &= \left[Y_{O_{inst}} \cdot \frac{f_y}{H_{comp}} \right] \\
 x_{lim} &= \left[\left(X_{O_{inst}} + W_{t_{inst}} \right) \cdot \frac{f_x}{W_{comp}} \right]; & y_{lim} &= \left[\left(Y_{O_{inst}} + H_{t_{inst}} \right) \cdot \frac{f_y}{H_{comp}} \right]
 \end{aligned} \tag{C-3}$$

Cabe observar que es posible que la región de ventana seleccionada modificada, definida por s_x'' , s_y'' , o_x'' y o_y'' , puede caer ligeramente a la izquierda o por encima del origen, en otras palabras, o_x'' y/o o_y'' pueden ser negativos. Toda porción de la región de imagen seleccionada que se encuentre a la izquierda o por encima del origen se ignorará al determinar la región de imagen de tren de código, conforme al procedimiento descrito en C.4.1.

Si no se suministran los valores `jpx-isset` y `jpx-inum`, los parámetros de región modificada que se han de utilizar en lugar de f_x , f_y , s_x , s_y , o_x y o_y se calculan utilizando las expresiones de la ecuación C-4. Como antes, estos parámetros modificados se utilizan al establecer la resolución y región de imagen de tren de código, conforme al procedimiento de C.4.1.

$$\begin{aligned}
 f_x'' &= \left[f_x \cdot \frac{X_{R_{reg}}}{X_{S_{reg}}} \cdot \frac{W_{cod}}{W_{reg}} \right]; & f_y'' &= \left[f_y \cdot \frac{Y_{R_{reg}}}{Y_{S_{reg}}} \cdot \frac{H_{cod}}{H_{reg}} \right] \\
 o_x'' &= o_x - \left[\frac{X_{O_{reg}}}{X_{S_{reg}}} \cdot \frac{f_x}{W_{reg}} \right]; & o_y'' &= o_y - \left[\frac{Y_{O_{reg}}}{Y_{S_{reg}}} \cdot \frac{f_y}{H_{reg}} \right] \\
 s_x'' &= s_x; & s_y'' &= s_y
 \end{aligned} \tag{C-4}$$

El segundo tipo de `context-range` descrito en esta Recomendación | Norma Internacional, `mj2t-context`, permite a los clientes solicitar pistas específicas de un fichero MJ2. El identificador `mj2-track` tendrá que ser un entero positivo, puesto que 1 es el menor identificador permitido de pista dentro de uno de dichos ficheros. Cuando el identificador `mj2-track` incluya el subfijo facultativo "+now" el `mj2t-context` estará compuesto por todos los trenes de código que forman parte de la pista de vídeo MJ2, empezando con aquel cuyo tiempo de captura corresponde al momento de recepción de la petición, algo útil cuando la fuente sea un tren de vídeo en directo. De lo contrario, el servidor puede asociar "now" con cualquier tren de código que le parezca adecuado. Si no se incluye el subfijo "+now", el `mj2-context` contendrá todos los trenes de código que pertenezcan a la pista de vídeo MJ2.

Un `mj2t-context` puede especificar una transformación de recorrespondencia de coordenadas que se ha de utilizar al determinar las resoluciones y regiones de imagen de tren de código para cada uno de sus trenes de código. De no haberlo, se interpretarán los parámetros de tamaño de cuadro y región suministrados a través de los campos de petición tamaño de cuadro, desplazamiento y tamaño de región conforme al procedimiento descrito en C.4.1. De lo contrario, se está solicitando uno de los dos tipos de transformación de coordenadas, tal como se identifica mediante la aparición de uno de los testigos "track" o "movie".

ISO/CEI 15444-9:2005 (S)

Cuando se especifica "track", indica que se están utilizando los campos de petición tamaño de cuadro, desplazamiento y tamaño de región para identificar un tamaño de presentación y una región rectangular deseada dentro del menor rectángulo limitante que contenga la presentación de las pistas, y dentro de este tamaño de presentación deseado. Se aplicará la transformación geométrica descrita por la casilla encabezamiento de pista MJ2 (tkhd) para establecer una resolución y región de imagen correspondiente en cada tren de código asociado con la pista.

Cuando se especifique "movie", indica que se están utilizando los campos de petición tamaño de cuadro, desplazamiento y tamaño de región para identificar un tamaño deseado para toda la película reproducida (que puede ser compuesta), y una región rectangular deseada dentro del rectángulo menor de limitación que contenga la película, a este tamaño determinado. Se combinarán las transformaciones geométricas descritas por la casilla de encabezamiento de pista MJ2 (tkhd) con las transformaciones geométricas descritas por la casilla encabezamiento de película (mvhd) y el resultado se utilizará para establecer una resolución y región de imagen correspondiente en cada tren de código asociado con la pista.

Cuando el servidor no pueda aplicar ninguna de las transformaciones geométricas `mj2t-context` descritas, proveerá un `mj2t-context` modificado en su encabezamiento de respuesta de contexto de tren de código.

NOTA 1 – Es posible que al utilizar el campo petición de contexto de tren de código junto con el campo petición tren de código se solicite múltiples veces un tren de código con diferentes transformaciones geométricas de los campos de petición tamaño de cuadro, tamaño de región y desplazamiento. Cuando esto ocurra, se estará pidiendo múltiples porciones de imagen no conexas o superpuestas de dicho tren de código.

NOTA 2 – Es posible obtener un resultado equivalente al de la ecuación C-4 mediante la transformación $X_{S_{comp}}=W_{S_{inst}}=W_{t_{inst}}=W_{reg}$, $Y_{S_{comp}}=H_{S_{inst}}=H_{t_{inst}}=H_{reg}$ y $XO_{inst}=YO_{inst}$ en la ecuación C-3, cuando los límites de s_x , s_y , o_x y o_y no estén limitados por X_{lim} , X_{min} , Y_{lim} , Y_{min} .

EJEMPLO 1: "context=jpxl<0-4:2>[s5i2]"

En este caso, se solicita al servidor devolver los trenes de código que están siendo utilizados por las capas de composición JPX 0, 2 y 4, haciendo una recorrespondencia del tamaño de cuadro y región de imagen solicitados conforme a los ajustes geométricos representados por la tercera instrucción de la sexta casilla de conjunto de instrucciones dentro de la casilla de composición (en los ficheros JPX hay por lo menos una casilla de composición).

EJEMPLO 2: "stream=0&context=mj2t<1+now>[track]"

En este caso, se solicita al servidor devolver el tren de código 0, así como todos los trenes de código que pertenezcan a la primera pista de un fichero MJ2, empezando desde aquel cuyo tiempo de muestreo sea el actual. Más aún, se pide al servidor hacer recorresponder el tamaño de trama y la región de imagen solicitados conforme a los ajustes geométricos descritos en la casilla encabezamiento de pista, descartando todo otro ajuste geométrico que puede haber sido descrito en la casilla encabezamiento de película.

C.4.8 Velocidad de muestreo (*srate*, o *sampling rate*)

```
srate = "srate" "=" streams-per-second
```

```
streams-per-second = UFLOAT
```

Si se suministra este campo, los trenes de código que pertenecen a la ventana seleccionada se obtienen mediante el submuestreo de aquéllos mencionados por el campo de petición tren de código, además de aquéllos ampliados a partir de los valores de context-range en el campo de petición contexto de tren de código (véase C.4.7), garantizando así que la velocidad promedio de muestreo sea inferior al valor de trenes por segundo. Esto es posible solamente si los trenes de código tienen información de temporización asociada (por ejemplo, si pertenecen a un objetivo lógico que sea conforme al formato de fichero MJ2).

Este campo de petición es útil solamente para establecer qué trenes de código se han de considerar como parte de la ventana seleccionada. El servidor podrá hacer un barrido a través de todos los trenes de código que normalmente se incluirían en la ventana seleccionada, para descartar aquellos que permitan garantizar que la separación promedio entre los tiempos de fuente de tren de código no sea menor que el valor recíproco de trenes por segundo. En esta Recomendación | Norma Internacional no se prescribe ningún algoritmo de submuestreo o ninguna interpretación precisa de la expresión "separación promedio".

De no haber información de temporización de fuente, la ventana seleccionada estará compuesta de todos los trenes de código identificados mediante los campos de petición tren de código y contexto de tren de código aunque este campo de petición puede de todos modos aceptar la interpretación de un campo de petición de velocidad de entrega, si lo hubiere.

C.4.9 Región de interés (*roi*)

```
roi = "roi" "=" region-name
```

```
region-name = 1*(DIGIT / ALPHA / "_")  
/ "dynamic"
```

Este campo especifica la región espacial deseada de la imagen utilizando un nombre en lugar de sus coordenadas. La correspondencia entre `region-name` y una determinada región espacial de la imagen puede tener varias formas, por ejemplo estar definida dentro de una casilla de descripción ROI dentro del objetivo lógico o dentro de la implementación del servidor propiamente dicho.

Un valor de `region-name` "dinámico" (una ROI dinámica) se reserva para representar una región no constante dentro de la imagen que se hace corresponder independientemente para cada petición a una región espacial. El servidor puede utilizar cualquier información acerca del cliente y cualquier otro parámetro de petición para determinar qué región espacial habrá de proporcionar como respuesta a determinada petición. Por ejemplo, si el servidor sabe que la pantalla del cliente es muy pequeña, puede decidir que sólo es necesaria una mayor resolución para el primer plano de la imagen mientras que la región entera de ésta se puede proporcionar con una resolución inferior. No es obligatorio que los servidores soporten las ROI dinámicas.

De haberlo, y cuando el servidor sea capaz de tratar peticiones ROI, el campo ROI tiene precedencia sobre los campos traslación y tamaño de región, en cuyo caso el servidor deberá ignorar estos últimos. Cuando hay ROI, pero el servidor no sabe cómo tratarlo, éste ignorará el campo ROI y utilizará los campos traslación y tamaño de región. Cuando estos campos no estén presentes, se utilizarán sus valores por defecto.

Si el cliente especifica un tamaño de cuadro así como una ROI, y el servidor está en condición de tratar la ROI especificada, se determina la resolución de imagen a la cual se pide la ROI utilizando el campo de petición tamaño de cuadro.

C.4.10 Capas (*layers*)

```
layers = "layers" "=" UINT
```

Se puede utilizar este campo para restringir el número de capas de calidad de tren de código que pertenecen a la petición de ventanas seleccionadas. Todas las capas disponibles son, por defecto, de interés. El valor especifica la cantidad de capas de calidad que interesan inicialmente. El servidor no intentará aumentar ningún bins de datos de recinto más allá de la frontera pertinente de capa. El servidor no intentará aumentar ningún bins de datos de losa más allá del punto para el cual todos los contenidos restantes pertenecen a la frontera de capa pertinente. Debido al orden de los datos dentro de una losa, es posible que el servidor deba retornar datos más allá de la frontera de la capa solicitada solamente para peticiones de tren JPT.

C.5 Campos de petición de metadatos

C.5.1 Metadatos solicitados implícitamente dentro de peticiones de ventana seleccionada

Los campos de petición de tren de código y de petición de contexto de tren de código identifican uno o varios trenes de códigos asociados con la ventana seleccionada solicitada. Aún si no hay ninguno de estos campos de petición, la ventana seleccionada corresponde a por lo menos un tren de código, como se menciona en C.4.6. Más aún, conforme a C.4.2 incluso si se omite el campo de petición de tamaño de cuadro, la ventana seleccionada solicitada incluye por lo menos el encabezamiento principal de tren de código para cada tren de código solicitado. La única excepción ocurre cuando se especifica `metadata-only` en un campo de petición de metadatos (véase C.5.2). Salvo en dicho caso, el cliente está solicitando implícitamente todas las casillas de metadatos que puedan ser necesarias conforme al formato de fichero, si lo hubiere para utilizar la imagen representada por los trenes de código solicitados. Con el fin de garantizar el interfuncionamiento entre las componentes de cliente y servidor, en esta subcláusula se identifica un conjunto mínimo de metadatos que los servidores habrán de considerar como solicitados implícitamente junto con la ventana seleccionada. Siempre que el servidor sepa de otros elementos pertinentes de metadatos, también puede entregarlos.

Para los ficheros JP2 y JPX, se deben considerar como pedidos los siguientes elementos de metadatos junto con la ventana seleccionados.

- a) El contenido completo del bin de metadatos 0.
- b) El contenido completo de cada una de las casillas siguientes, siempre que se encuentren en la parte superior del fichero:
 - 1) Firma JP2 ("jP").
 - 2) Tipo de fichero ("ftyp").
 - 3) Requisitos de lector ("req").
 - 4) Composición ("comp").

- c) Todo los encabezamientos inmediatos de subcasilla de cada una de las siguientes supercasillas:
 - 1) Toda casilla de encabezamiento ("jp2h").
 - 2) Toda casilla de encabezamiento de tren de código ("jpch") asociada con un tren de código solicitado.
 - 3) Toda casilla de encabezamiento de capa de composición ("jplh") asociada con una capa de composición JPX solicitada a través del campo de petición contexto de tren de código.
- d) El contenido completo de cada una de las siguientes casillas, siempre que se encuentren dentro de una de las supercasillas mencionadas:
 - 1) Encabezamiento de imagen ("ihdr").
 - 2) Bits por componente ("bpc").
 - 3) Palette ("pclr").
 - 4) Correspondencia de componente ("emap").
 - 5) Definición de canal ("cdef").
 - 6) Resolución ("res").
 - 7) Registro de tren de código ("crg").
 - 8) Opacidad ("opct").
- e) Para ficheros JP2, compatibles con JP2 y JPX, una o varias de las casillas de descripción de espacio de color ("colr") asociadas con cada tren de código o capa de composición JPX pedida a través del campo de petición con texto de tren de código, a saber:
 - 1) Si el servidor puede determinar exactamente qué casilla es la preferida, habrá de enviar solamente dicha casilla, aún si esto implica no enviar la primera casilla para los ficheros JP2 o compatibles JP2 (por ejemplo, si la segunda casilla es Any ICC, y se especifica que el cliente prefiere Any ICC). Si el servidor no puede determinar con exactitud la casilla preferida, habrá de enviar la primera casilla descripción de espacio de color completa.
 - 2) Para todas las casillas que no se han enviado, el servidor enviará una porción del contenido de casillas, de tal manera que el cliente pueda establecer si desea más adelante pedir otra especificación de espacio de color.
 - Para las casillas enumeradas, el servidor tendrá que enviar por lo menos los primeros 7 bytes del contenido de la casilla (hasta como mínimo el campo EnumCS).
 - Para las casillas de espacio de color definidas por el fabricante, el servidor enviará por lo menos los primeros 19 bytes del contenido de casilla (hasta como mínimo el campo VCLR).
 - Para las casillas de espacio de color Restricted y Any ICC, el servidor habrá de enviar por lo menos los primeros 3 bytes del contenido de casilla (al menos los campos METH, APPROX y PREC).

Se solicita al servidor retornar un prefijo inicial para cada bin de metadatos que contenga cualquiera de los anteriormente mencionados, que vaya desde el primer byte de los bin de metadatos y continúe hasta el final de todos los bins de metadatos solicitados de dicho bin de metadatos. Siendo así, la cantidad total de metadatos que retorna el servidor puede depender de cómo se haya dividido el objetivo lógico en bins de metadatos. En A.3.6.2 se presentan más detalles al respecto.

Si bien en esta Recomendación | Norma Internacional no se trata la composición de los metadatos MJ2 implícitos para las peticiones de ventana seleccionada, es posible que sea objeto de una norma futura.

C.5.2 Petición de Metadatos (*metareq*)

```

metareq = "metareq" "=" 1#("[" 1$(req-box-prop) "]" [root-bin] [max-depth])
          [metadata-only]

req-box-prop = box-type [limit] [metareq-qualifier] [priority]

limit = ":" (UINT / "r")

metareq-qualifier = "/" 1*("w" / "s" / "g" / "a")

priority = "!"

root-bin = "R" UINT
    
```



```
max-depth = "D" UINT
```

```
metadata-only = "!!"
```

Este campo especifica qué metadatos se desean como respuesta a la petición, además de todos los metadatos requeridos por el cliente para decodificar o interpretar los datos de imagen solicitada (véase C.5.1). La cadena valor de este campo petición es una lista de peticiones independientes; no obstante, el servidor puede tratarlas como un grupo, y es posible entonces que se superpongan entre peticiones.

Cada petición se relaciona con el bin de datos especificado por su valor `root-bin`. Si no se especifica dicho valor, la raíz es el bin de metadatos 0. La petición se refiere solamente a los datos que pertenecen a dicho bin de datos o a los que se hace referencia en él (mediante las casillas marcador de posición).

Si se especifica un valor para `max-depth`, sólo se están pidiendo las casillas contenidas dentro del bin de metadatos raíz y aquellas que no estén por debajo de los niveles `max-depth` en la jerarquía de fichero. De no haberlo, quiere decir que no hay límite en la profundidad de la jerarquía de ficheros para esta petición.

La porción `req-box-prop` de la petición especifica una lista de tipos de casilla que no interesan al cliente. La cadena especial "*" se puede sustituir por el tipo de casilla, en cuyo caso se refiere a todos los tipos de casilla. Cada tipo de casilla (o "*") puede ir seguido de cualquier combinación de los tres atributos: un valor límite, un calificador `metareq` y una bandera de prioridad.

`limit` especifica qué tipo de información y qué parte del contenido de la casilla está solicitando el cliente para dicho tipo de casilla. El parámetro límite se presenta como dos puntos seguidos por un valor (el valor límite) que habrá de ser un entero sin signo o el carácter "r".

Si el valor límite es un entero n mayor que cero, se solicita al servidor retornar solamente los primeros n bytes del contenido de las casillas pertinentes de dicho tipo de casilla, además de los encabezamientos de casilla. Si el valor límite es el número 0, se están solicitando solamente los encabezamientos de casilla para las casillas de dicho tipo. Si no se especifica `limit`, el cliente está solicitando el contenido entero de todas las casillas de todo tipo que satisfagan los demás requisitos de la petición, sin importar que las casillas de dicho tipo sean supercasillas o no. De igual manera, de haber un valor numérico o no especificado del límite en una supercasilla, se estará solicitando al servidor proporcionar la cantidad de datos pedida por `limit` sin importar si la jerarquía contenida dentro de la casilla es más profunda que la que se habría alcanzado sobre la base de los valores `root-bin` y `max-depth`, e independientemente de los tipos de casillas o subcasillas que se encuentran dentro de la supercasilla.

Si el valor límite es "r" se solicita al servidor enviar el encabezamiento de casilla, más no el contenido, para toda casilla dentro del tipo de casilla indicado, así como para todas sus subcasillas vástagos (sin importar su tipo de casilla), hasta la profundidad máxima especificada en la petición. Siendo así, esto corresponde a una petición de la estructura de dicha porción en la jerarquía de casillas. Cuando el servidor no pueda establecer si una casilla es supercasilla, no debería ser capaz de acceder recursivamente en la subcasilla de la casilla, lo que impediría responder completamente a algunas de las peticiones de metadatos. Los servidores han de poder reconocer el estatus de supercasilla de todas las casillas definidas para todos los formatos de ficheros que se prevé soporten.

Mientras que un valor límite "r" implica que el cliente está solicitando el esqueleto de la estructura de casilla, que está compuesto por los encabezamientos de casilla, puede ocurrir que debido a la división del objetivo lógico en bins de metadatos el servidor se vea forzado a devolver otros datos, incluido en contenido de algunas casillas y los encabezamientos y/o contenido de otras casillas que no hayan sido solicitadas. Esto se debe a que se pide al servidor devolver todos los bytes desde el inicio de cada bin de metadatos que contenga bytes de casilla solicitada hasta el último byte de casilla solicitada.

El `metareq-qualifier` tiene la forma de "/" seguido por una o varias de las banderas "g", "s", "w" y "a", cada una de las cuales identifica un contexto a partir del cual se han de tomar las casillas que corresponden a la petición. En el cuadro C.2 se indica la interpretación de cada una de ellas. Si se suministra más de una de las banderas, se ha de tomar la unión de los contextos correspondientes. Si no se proporciona `metareq-qualifier`, se habrá de utilizar la unión de "g", "s" y "w". A efectos de una mayor claridad, conviene observar que los contextos "g", "s" y "w" se excluyen mutuamente pero su unión es en general menor que el contexto "a" que incluye todo.

Si se especifica la bandera `priority`, el cliente está solicitando que las casillas del tipo `box-type` que correspondan a otros elementos de la petición se entreguen con más prioridad que los datos de imagen.

Si un tipo de casilla no se especifica en la lista `req-box-prop`, significa que no se están solicitando datos para casillas de dicho tipo.

Si se especifica `metadata-only` al final del campo de petición de metadatos, el cliente solicita que la respuesta del servidor conste solamente de metadatos, sin que haya datos de imagen o encabezamientos de tren de código, independientemente si se han utilizado campos de petición de ventana seleccionada como por ejemplo tamaño de

ISO/CEI 15444-9:2005 (S)

cuadro. En el caso de los tipos de retorno de tren JPP y JPT, esto implica que los mensajes JPIP devueltos serán todos mensajes de bin de metadatos.

EJEMPLO 1: "metareq=[*]R31D4"

En este caso, se solicita al servidor devolver el contenido completo de todas las casillas que encuentre en el contenido del bin 31. Si bien se ha señalado una restricción en la profundidad deseada, el servidor la ignorará puesto que el contenido de aquellas casillas no se limitó mediante el parámetro `limit`.

EJEMPLO 2: "metareq=[*:r,drep]R31D4"

"*:r" implica que se ha solicitado al servidor devolver encabezamientos de casilla para todas las casillas contenidas en el bin de metadatos 31 y cualquiera de los bins a los que se hace referencia mediante los marcadores de posición incluidos dentro de dicho bin, hasta una profundidad de 4 niveles del contenido del bin 31, pero sin incluir el contenido de dichas casillas. La `req-box-prop` "drep" adicional especifica que se pide al servidor devolver el contenido completo de toda casilla "drep" incluida dentro del bin de metadatos 31 y cualquier de los bins a los que se haga referencia mediante marcadores de posición dentro de dicho bin, hasta una profundidad de 4 niveles del contenido del bin 31.

EJEMPLO 3: "metareq=[drep]R31D4"

En este caso, se solicita aún al servidor retornar el contenido completo de toda casilla "drep" que encuentre en el contenido del bin 31 o cualquier de los bins a los que se haga referencia mediante dicho bin, hasta una profundidad de cuatro niveles del contenido del bin 31. No obstante, dado que no se especificaron otras casillas, se solicita al servidor enviar solamente tantos otros datos como sea necesario para especificar la posición de toda casilla "drep" en la jerarquía de fichero con respecto a la casilla contenida dentro del bin de metadatos 31.

Independientemente de las especificaciones de casilla que se suministren a través del campo Petición de Metadatos, el servidor puede enviar más información, bien sea porque ha establecido que ésta es necesaria para que el cliente codifique o interprete los datos de imagen solicitada o porque el servidor ha dividido previamente el objetivo lógico en bins de datos basándose en criterios diferentes por lo que se han de enviar datos adicionales a fin de permitir una visión coherente y con significado de los bins de metadatos para dicho objetivo lógico.

Cuadro C.2 – Banderas calificador de petición de metadatos

Bandera	Interpretación
"w"	Este contexto metareq incluye todas las casillas que se sabe que están asociadas con determinada región de imagen espacial dentro de uno o varios trenes de código que pertenecen a la ventana seleccionada, y donde la región espacial, las resoluciones y las componentes de imagen con las que se relaciona la casilla se interceptan con las de la ventana seleccionada. Se ha de establecer, por ejemplo, este tipo de asociación mediante una casilla "asoc" en un fichero JPX.
"s"	Este contexto metareq incluye todas las casillas que se sabe que se deben asociar con uno o varios trenes de código que pertenecen a la ventana seleccionada, o con uno o varios de los contextos de tren de código solicitados (por ejemplo, capas de composición JPX o pistas de vídeo MJ2), y donde dichas casillas no se asocian solamente con regiones espaciales determinadas. Se puede establecer, por ejemplo, dicha asociación mediante una casilla "asoc" en un fichero JPX.
"g"	Este contexto metareq incluye todas las casillas que son del caso en la ventana seleccionada solicitada, teniendo en cuenta los trenes de código solicitados y los contextos de tren de código solicitado, salvo por aquellas casillas que se incluyen en los contextos metareq "w" y "s".
"a"	Este contexto metareq incluye todas las casillas en el objetivo lógico, sin excepción (nota).

NOTA – Este contexto metareq es adecuado para las peticiones en las que se desea la estructura de fichero independientemente de la ventana seleccionada.

C.6 Campos de petición de limitación de datos

C.6.1 Máxima longitud de respuesta (*len*)

`len = "len" "=" UINT`

Este campo especifica una restricción en la cantidad de datos que el cliente desea recibir del servidor en respuesta a su petición. Ha de tener como unidad el byte. De no haberlo, el servidor deberá enviar datos de imagen al cliente hasta que parezca que se han enviado todos los datos del caso, se alcance un límite de calidad (véase C.6.2) o la respuesta haya sido interrumpida por la llegada de una nueva petición que no incluye un campo de petición esperar (`wait`) cuyo valor sea "yes" (véase C.7.2). El cliente puede utilizar `len=0` cuando solicite encabezamientos de respuesta mas no datos de respuesta.

C.6.2 Calidad (*quality*)

```
quality = "quality" "=" (1*2DIGIT / "100") ; 0 to 100
```

Este campo se puede utilizar para limitar la calidad de transmisión de datos a un cierto nivel (entre 0 para la calidad menor y 100 para la mayor) asociado con la imagen. Es difícil establecer límites de calidad de una manera fiable, y el servidor puede ignorar esta petición respondiendo con un valor "-1" (véase D.2.16). No obstante, es útil permitir al cliente dar alguna indicación de la máxima calidad de imagen que puede ser de su interés. El factor de calidad puede tratar de aproximarse a la calidad ad hoc que se suele utilizar para el control de la compresión JPEG. El cliente ha de esperar que el tamaño de datos devueltos sea monótonamente no decreciente al aumentar la calidad, es decir, un aumento en el valor de la calidad suele corresponder a un aumento del tamaño de datos devuelto.

NOTA – Si un servidor soporta esta petición y dos clientes diferentes hacen peticiones idénticas relativas al mismo objetivo con el mismo valor de calidad, es decir "quality=80", el servidor ha de tener una política coherente de implementación al devolver datos de los bins de datos.

C.7 Campos de petición de control de servidor**C.7.1 Alineamiento (*align*)**

```
align = "align" "=" ("yes" / "no")
```

Este campo especifica si los datos de respuesta del servidor han de alinearse con las fronteras naturales. El valor por defecto es "no". Si el valor es "yes" todo mensaje de tren JPT o JPP entregado como respuesta a esta petición y que cruce una "frontera natural" será terminado en cualquier "frontera natural" subsiguiente. En el cuadro C.3 se enumeran las fronteras naturales para cada tipo de bin de datos. Se dice que un mensaje cruza una frontera natural cuando incluye el último byte antes de la frontera, y el primero después de ella. Por ejemplo, un bin de datos de recinto cruza una frontera natural si incluye el último byte de un paquete y el primero del próximo. Obsérvese que en realidad no se requiere que los mensajes de respuesta alineados terminen en una frontera natural a menos que la crucen. Esto quiere decir, por ejemplo, que se pueden incluir en la respuesta paquetes parciales de recintos, que han de ser necesarios cuando un límite de bytes prevaleciente impida la entrega de los paquetes completos.

Cuadro C.3 – Fronteras de alineamiento basadas en el tipo de bin

Tipo de bin	Frontera natural
Bin de datos de recinto	Fin de un paquete (una frontera para cada capa de calidad)
Bin de datos de loza	Fin de una parte losa (una frontera para cada parte losa)
Bin de datos de encabezamiento de loza	Fin del bin (una sola frontera)
Bin de datos de encabezamiento principal	Fin del bin (una sola frontera)
Bin de metadatos	Fin de una casilla en el nivel superior del bin de datos (una frontera para cada casilla)

C.7.2 Espera (*wait*)

```
wait = "wait" "=" ("yes" / "no")
```

Este campo se utiliza para indicar si el servidor debe completar una respuesta a una petición anterior. Si el valor es "yes", el servidor ha de responder completamente a la petición anterior en el mismo canal especificado a través del campo ID de canal antes de empezar a responder la petición.

Si el valor es "no", el servidor puede terminar paulatinamente el tratamiento de toda petición previa en el mismo recurso de canal (especificado a través del campo ID de canal) antes de completarlo y puede empezar a responder a esta nueva petición. En este contexto, la "terminación paulatina" implica que el servidor debe completar por lo menos el mensaje en curso.

El valor por defecto de este campo es "no".

C.7.3 Tipo de retorno de imagen (*type*)

```
type = "type" "=" 1#image-return-type
image-return-type = media-type / reserved-image-return-type
media-type = TOKEN "/" TOKEN *( ";" parameter )
reserved-image-return-type = TOKEN *( ";" parameter )
parameter = attribute "=" value
```

ISO/CEI 15444-9:2005 (S)

attribute = TOKEN

value = TOKEN

Este campo se utiliza para indicar el tipo (o tipos) de los datos de respuesta solicitados. Un servidor que no desee proporcionar alguno de los tipos de retorno solicitados tendrá que emitir una respuesta de error.

El valor del campo petición tipo de retorno de imagen deberá ser bien un tipo de medios (definido en RFC 2046) o bien uno de los tipos de retorno de imagen reservados que se define en el cuadro C.4.

Cuadro C.4 – Tipos válidos de retorno de imagen

Tipo	Interpretación
"jpp-stream"	Un tren JPP como se define en el anexo A. Un "jpp-stream" puede facultativamente ir seguido de un ";ptype=ext", en cuyo caso el tipo de retorno solicitado es uno en el cual todos los encabezamientos de mensaje de bin de datos de recinto tienen la forma extendida. (véase A.2.2)
"jpt-stream"	En el anexo A se define un tren JPT. "jpt-stream" puede facultativamente ir seguido por un ";ttype=ext" en cuyo caso el tipo de retorno es uno en el que todos los encabezamientos de mensaje de bin de datos de loza tienen la forma extendida (véase A.2.2)
"raw"	El cliente está solicitando que se entregue sin cambios la secuencia entera de bytes en el objetivo lógico.
Otros valores	Reservado para la utilización por parte de la ISO.

Si se omite el campo de petición `ttype`, el tipo de retorno habrá de establecerse por otros medios.

En una sesión, es decir cuando las peticiones involucren un campo petición ID de canal, el valor del parámetro de retorno se mantendrá en las respuestas sucesivas para las peticiones de datos o metadatos de imagen que corresponda al mismo objetivo lógico.

NOTA 1 – Un servidor puede proporcionar otros tipos de medios de imagen (por ejemplo, jp2, jpeg, tiff, png), de haberlos, como un servicio de transcodificación con funcionalidad JPIP.

NOTA 2 – Para el caso del tipo de retorno de tren de código bruto, los datos de respuesta tienen que estar compuestos por la entidad solicitada completa. Por tanto, muchos de los otros campos de petición de cliente que son posibles podrían no tener significado y deberían en tal caso ser ignorados por el servidor.

C.7.4 Velocidad de entrega (*drate*)

drate = "drate" "=" rate-factor

rate-factor = UFLOAT

Este campo se utiliza para especificar la velocidad de entrega de diversos trenes de código. Si se suministra este campo, el servidor entregará los datos pertenecientes a los diversos trenes de código en la ventana seleccionada que viene después de una planificación secuenciada temporalmente. Los trenes de código que pertenecen a la ventana seleccionada son todos aquellos que se identifican mediante un campo petición de tren de código y el campo petición de contexto de tren de código, que han sido tal vez submuestreados conforme al campo petición velocidad de muestreo.

A fin de conferir un significado a este campo de petición, se debe asociar información de temporización con los diversos trenes de código en la ventana seleccionada. Si los trenes de código forman parte de un fichero MJ2, dicho fichero proporciona la información de temporización y permite una correspondencia entre cada tren de código y un tiempo de ejecución nominal, que se identifica en adelante como "el tiempo de origen".

Si los trenes de código no poseen información de temporización de origen, pero si hay un campo de petición de velocidad de muestreo, el servidor habrá de suponer que los trenes de código de la ventana seleccionada tienen tiempos de origen separados por un valor equivalente al recíproco del valor del campo petición de velocidad de muestreo.

Si los trenes de código no poseen información de temporización de origen, y no está presente el campo de petición de velocidad de muestreo, el servidor tendrá que suponer que los tiempos de código de la ventana seleccionada tienen tiempos de origen separados por exactamente un segundo.

El campo de petición de velocidad de entrega suministra un factor de escala entre las velocidades de entrega y origen. Si viene dado por 1, el servidor deberá intentar la entrega de los trenes de código al cliente a la velocidad sugerida por sus tiempos de origen, observando eso si que dichos tiempos de origen no tienen por que necesariamente ser regulares. En general, si el factor de velocidad es F , el servidor deberá tratar de entregar los trenes de código al cliente a una velocidad que sea F veces mayor que la sugerida por sus tiempos de origen.

Si el servidor no puede entregar todos los datos pertinentes para cada tren de código a la velocidad solicitada (por ejemplo, debido a restricciones de ancho de banda), deberá entregar solamente parte de los datos para cada tren de código, de tal manera que no se viole la velocidad de entrega solicitada. La porción de cada tren de datos de tren de

código que no se entrega puede depender del valor `view-window-pref` suministrado en un campo de petición preferencias de cliente (véase C.10.2). Si la preferencia es "progresiva" o si no se identifica dicha preferencia, el servidor tratará de entregar una calidad de imagen uniforme y máxima en la ventana seleccionada, sujeto a la restricción en velocidad de entrega. Si se ha suministrado un valor `view-window-pref` de "fullwindow", el servidor tendrá que, de alguna manera, cortar la representación asociada con cada tren de código. En todo caso, el comportamiento ha de ser similar al que se habría presentado en el caso de un cliente que emita una sucesión de peticiones para cada uno de los trenes pertinentes, a la velocidad de entrega.

Si el servidor puede entregar todos los datos del caso para cada tren de código y a la velocidad solicitada, deberá mantener la conexión en reposo cuando sea necesario para poder garantizar que no se supere la velocidad de entrega.

Si no se suministra este campo y si no ha sido especificado un valor `view-window-pref` de "fullwindow", el servidor deberá tratar de poner en una secuencia los datos pertinentes de tal manera que se incremente progresivamente la calidad de todos los trenes de código de una manera uniforme.

C.8 Campos de petición de gestión de caché

C.8.1 Modelo (*model*)

C.8.1.1 Generalidades

```

model = "model" "=" 1#model-item
model-item = [codestream-qualifier ","] model-element
model-element = ["-"] bin-descriptor
bin-descriptor = explicit-bin-descriptor ; C.8.1.2
                / implicit-bin-descriptor ; C.8.1.3
codestream-qualifier = "[" 1$(codestream-range) "]"
codestream-range = first-codestream-id ["-"] [last-codestream-id]
first-codestream-id = UINT
last-codestream-id = UINT

```

Este campo se puede utilizar en peticiones basadas en sesión o sin estado. Una petición basada en sesión es aquella que incluye un campo Channel-ID, puesto que se hacen corresponder canales con una sesión gestionada por un servidor. El campo "modelo" contiene uno o varios descriptores de bins, cada uno de los cuales identifica un bin de datos, acerca de la cual se está señalando información de caché. Para las peticiones dentro de una sesión, esta información caché sirve para actualizar el modelo del servidor de la caché del cliente. Hay un sólo modelo caché para cada objetivo lógico asociado con la sesión. En el caso de una petición sin estado, el modelo del servidor de la caché de cliente está vacío al inicio de la petición, pero se actualiza gracias al campo "modelo" (si lo hay) antes de que el servidor emita su respuesta. Cuando el servidor finaliza el tratamiento de una petición sin estado se descarta toda la información del modelo caché.

Se suministran dos formas de valores de descriptor de bin para facilitar el intercambio eficaz de información de modelo caché, a saber, la forma "explícita" y la "implícita", que se describen en la subcláusula siguiente. Los clientes pueden emitir peticiones utilizando una u otra forma, o una mezcla de las dos dentro de un solo campo de petición "modelo", si fuera necesario.

Cuando un descriptor de bin viene precedido por un símbolo "-", se dice que es de sustracción. De lo contrario, se denomina de adición. Un descriptor de bins de sustracción informa al servidor que los datos pertinentes se han de remover del modelo de servidor de la caché de cliente. Por supresión de elementos de modelo de caché se entiende que el servidor no podrá suponer que el cliente ya los posee. Los valores de descriptor de bin se tramitan en orden.

Un descriptor de bins de adición (uno que no esté precedido por el símbolo "-") informa al servidor sobre datos que ya están en la caché de cliente. Por tanto, el servidor puede añadir esta información a su modelo de caché y suponer que el cliente ya los posee.

El campo "modelo" puede hacer referencia a bins de datos que no sean pertinentes para la ventana seleccionada de interés identificada por otros campos de petición (Tamaño de cuadro, Tamaño de región, Offset, etc.), en cuyo caso la manipulación del modelo caché no debería afectar la respuesta a la petición en curso, aunque sí puede afectar la respuesta a futuras peticiones (a menos que se trate de una petición sin estado).

Siempre que la lista de ítems de modelo incluya un calificador de tren de código, todos los elementos de modelo subsiguientes se han de añadir o sustraer (cuando corresponda), de los trenes de código cuyos identificadores se listan mediante el calificador de tren de código. Se pueden intercalar en la lista los calificadores de tren de código con el fin

de cambiar progresivamente la recolección de trenes de código que puedan ser afectados por los elementos de modelo que vienen a continuación. Todo elemento de modelo que no vaya precedido por un calificador de tren de código puede ser el primer tren de código solicitado mediante un campo de petición de tren de código. De no haber este último campo, los valores de elemento de modelo que no vayan precedidos por un calificador de tren de código habrán de hacer referencia al tren de código 0, sin importar si se incluye o no un campo de petición de contexto de tren de código. Si no hay un last-codestream-id, pero sí está presente el guión del calificador, quiere decir que se incluye el first-codestream-id y todos los trenes de código subsiguientes.

Las peticiones dentro de una sesión no deben incluir ningún calificador de tren de código que haga referencia a más de un tren de código.

NOTA 1 – Si bien el servidor deberá tratar de aprovecharse de las declaraciones aditivas de manipulación de modelo caché, es libre de descartar algunas o todas de ellas para evitar posibles efectos en la eficacia del transporte. Conviene que los clientes sepan que es muy posible que un servidor descarte declaraciones aditivas de manipulación de modelos caché que se refieran a bins de datos que formen parte de trenes de códigos no afectados por la petición en curso. Para evitar la incertidumbre cuando se trate de múltiples trenes de código, se puede utilizar el campo de peticiones de "mset" para establecer el conjunto de trenes de código que se está modelando.

NOTA 2 – En general, la manipulación de un modelo de caché de un servidor basado en sesión influye tanto en la respuesta a la petición en curso como a cualquier petición futura. Más aún, todos los canales de una sesión asociados con un mismo objetivo lógico comparten el mismo modelo caché. Es decir, los modelos de las peticiones que lleguen a través de un canal (campo Channel ID) pueden afectar la respuesta a peticiones que lleguen a través de un canal diferente. Cabe observar que las peticiones que utilizan diferentes canales JPIP (otros valores del Channel ID) pueden llegar al servidor asincrónicamente, cuando se utilicen canales TCP independientes para transportar la petición bien sea directamente desde el cliente o indirectamente a través de un intermediario. Teniendo en cuenta estas consideraciones, los clientes han de tomar todas las medidas necesarias para garantizar el sentido de sus instrucciones de manipulación de modelo caché.

C.8.1.2 Forma explícita

```
explicit-bin-descriptor = explicit-bin
                        [ ":" (number-of-bytes / number-of-layers) ]

explicit-bin = codestream-main-header-bin
              / meta-bin
              / tile-bin
              / tile-header-bin
              / precinct-bin

number-of-bytes = UINT

number-of-layers = %x4c UINT ; "L"

codestream-main-header-bin = %x48 %x6d ; "Hm"

meta-bin = %x4d bin-uid ; "M"

tile-bin = %x54 bin-uid ; "T"

tile-header-bin = %x48 bin-uid ; "H"

precinct-bin = %x50 bin-uid ; "P"

bin-uid = UINT / "*"

```

Los valores del descriptor de bin que hacen referencia explícita a bins de datos son de uno de los tipos siguientes: M (bins de metadatos), Hm (bins de datos de encabezamiento principal), H (bins de datos de encabezamiento de losa), P (bins de datos de recinto) o T (bins de datos de losa). El o los bins de datos pertinentes dentro de los trenes de código del caso vienen identificados por descriptores de bins explícitos, bien sea utilizando un solo identificador de valor entero, o un carácter comodín, "*". La única excepción a esta regla es el bin de datos de encabezamiento principal de tren de código, cuyo descriptor de bin es (Hm). Para todas las otras clases de bins de datos, el identificador único es idéntico al valor comunicado en el identificador en clase en los encabezamientos de mensaje de tren JPP y/o JPT (véase el anexo A).

Se utilizará el carácter comodín, "*", solamente cuando se trate de peticiones sin estado, en cuyo caso el descriptor de bins se refiere simultáneamente a todos los bins de datos de la clase pertinente (metadatos, recinto, encabezamiento de losa o losa), que conciernen a la ventana seleccionada.

Se puede calificar cada descriptor de bin mediante una cierta cantidad de bytes. Un descriptor de bin aditivo que esté calificado mediante el número de bytes *B* indica que el cliente ya tiene en su caché por lo menos los primeros *B* bytes del bin de datos indicado; el servidor puede añadir los primeros *B* bytes del bin de datos a su modelo caché. Un descriptor de bin de sustracción que esté calificado mediante el número de bytes *B* indica que el cliente tiene a lo sumo

los primeros *B* bytes del bin de datos indicado; el servidor eliminará entonces de su modelo caché todo byte que venga después de los primeros *B* bytes del bin de datos.

EJEMPLO 1: Un descriptor de bin de sustracción calificado, como por ejemplo "-P23:10", indica que el servidor debe suprimir de su modelo caché todos los bytes, salvo los primeros del bin de datos 23 de recinto. Esto no implica que el cliente tenga los primeros 10 bytes del bin de datos 23 de recinto en su caché y el servidor no puede suponerlo añadiéndolos si no estuvieran ya presentes.

De otra parte, se pueden calificar los descriptores de bin de recinto mediante un número de capas. Un descriptor de bin de adición que esté calificado por el número de capas, *L*, indica que el cliente ya tiene por lo menos las primeras *L* capas (primeros *L* paquetes) del recinto indicado; el servidor puede añadir los bytes correspondientes a estas capas a su modelo caché. Un descriptor de bin de recinto de sustracción que esté calificado por el número de capas *L* indica que el cliente tiene a lo sumo las primeras *L* capas (*L* paquetes) del recinto indicado; el servidor deberá suprimir de su modelo caché los bytes correspondientes a toda capa subsiguiente de dicho recinto.

Un descriptor de bin que no posea calificador de número de bytes ni del número de capas se refiere a todo el bin de datos explícito.

EJEMPLO 2: "model=M0,Hm,H7:20,P3" significa que el cliente tiene en su caché por lo menos todos los bins de metadatos 0, todo el encabezamiento de tren de código principal, los primeros 20 bytes del encabezamiento de losa 7, y todo el recinto 3.

EJEMPLO 3: "model=P3:256,P5:L2,-P6:20" significa que el cliente tiene por lo menos los primeros 256 bytes del recinto 3 y las dos primeras capas (paquetes) del recinto 5, aunque (a lo sumo) no tiene nada más allá del vigésimo byte del recinto 6 (puede que tampoco tenga los primeros 20 bytes).

EJEMPLO 4: "model=M*,-M5,-H*,-P*:L3" indica que el cliente tiene (o está preparado para hacer creer al servidor que los tiene) todos los bins de metadatos salvo el bin de metadatos 5, ningún bin de datos de encabezamiento de losa que sea relevante para la ventana seleccionada y a lo sumo las primeras 3 capas de todo recinto es pertinente para la ventana seleccionada. Obsérvese, que se permiten los comodines utilizados cuando la declaración "modelo" aparece en una petición sin estado.

EJEMPLO 5: "model=[30-200],Hm,H*,M*,P0,[0-29],-Hm,-H*,-M*,-P*" significa que el cliente tiene todos los encabezamientos y metadatos, más el bin de datos de recinto 0 de los trenes de código en 30 a 200 incluidos, pero que ha suprimido todo encabezamiento, metadatos y bins de datos de recinto de los primeros en 30 trenes de código.

C.8.1.3 Forma implícita

```
implicit-bin-descriptor = 1*implicit-bin [":" number-of-layers]
implicit-bin = implicit-bin-prefix (data-uid / index-range-spec)
implicit-bin-prefix = %x74      ; t -- tile
                    / %x63      ; c -- component
                    / %x72      ; r -- resolution level
                    / %x70      ; p -- position
index-range-spec = first-index-pos "-" last-index-pos
first-index-pos = UINT
last-index-pos = UINT
data-uid = UINT / "*"

```

Los valores de descriptor de bin que hacen referencia implícitamente a los bins de datos son de uno de los siguientes tipos: *t* (loza a la cual pertenece el recinto), *c* (componente de imagen a la cual pertenece el recinto), *r* (nivel de resolución del componente de losa a la cual pertenece el recinto) o *p* (posición del recinto dentro de su resolución de componente de losa). Los descriptores de bin implícitos se utilizan para identificar bins de datos de recinto utilizando índices. Todos los índices deben empezar desde 0. Un índice de nivel de resolución igual a 0, *r*0, se refiere al menor nivel de resolución (subbanda LL) de la componente losa. Los índices de posición, *p*, van de izquierda a derecha y de arriba abajo en la progresión de la resolución de componente de losa, en barrido de línea, como se describe en la Rec. UIT-T T.800 | ISO/CEI 15444-1.

En las peticiones sin estado, se pueden reemplazar con el intervalo de índices o con el carácter comodín "*", cualquiera o todos los especificadores de bin implícito de losa, componente, nivel de resolución o posición. En cada caso, se expande el descriptor de bin para que incluya todos los valores de la gama de índices que atañe a la ventana seleccionada. Ninguna de estas opciones se debe utilizar para las peticiones dentro de una sesión.

ISO/CEI 15444-9:2005 (S)

En las peticiones sin estado, se puede también reemplazar por una gama única de índices cualquiera o todos los índices de losa, componente, nivel de resolución o posición. El valor first-index-pos en un index-range-spec proporciona el primer índice en una gama. El valor last-index-pos suministra el último índice en el intervalo y ha de ser mayor o igual que el valor del first-index-pos. Ambos índices especificados son inclusivos. No se puede omitir el last-index-pos. Cuando se dé un intervalo de índices de losa ("t"), el intervalo se refiere a una matriz rectangular de losas en cuya esquina superior izquierda va el valor de first-index-pos y en la inferior derecha el valor de last-index-pos. Del mismo modo, cuando se dé un intervalo de índice de posición ("p") el intervalo se refiere a una matriz rectangular de posiciones de recinto, en la que las esquinas superior izquierda e inferior derecha vienen dadas por los valores first-index-pos y last-index-pos, respectivamente. Tal como ocurre con los comodines, no se deben utilizar los intervalos cuando se trate de peticiones en una sesión.

Se pueden calificar los descriptores del bin de recinto implícitos mediante un número de capas, y en tal caso la sintaxis de interpretación son idénticas a las de los descriptores de bin de recinto explícitamente calificados, que se describen anteriormente.

EJEMPLO 1: "model=t0c2r3p4:L5" indica que el cliente tiene los primeros 5 paquetes del recinto 5 en orden, del cuarto nivel de resolución, de la tercera componente, de la losa 0.

EJEMPLO 2: "model=t10r0,t*r1:L4" significa que el cliente tiene todas las capas del índice de losa 10 al nivel de resolución 0, y las primeras 4 capas de todas las capas pertinentes para la ventana seleccionada a un nivel de resolución 1. Obsérvese que el comodín es adecuado solamente en el caso de peticiones sin estado.

EJEMPLO 3: "model=t0-10:L2" indica que el cliente tiene las primeras 2 capas de las losas 0 a 10. Obsérvese que el intervalo es adecuado solamente para las peticiones sin estado.

EJEMPLO 4: "model=t*r0-2:L4" indica que el cliente tiene las primeras 4 capas de los niveles de resolución 0 a 2 de todas las losas que tienen que ver con la ventana seleccionada. Obsérvese que tanto el comodín como el intervalo son adecuados solamente para las peticiones sin estado.

C.8.2 Resumen de las opciones de descriptor de caché (informativo)

Cuadro C.5 – Resumen de opción de descriptor de caché

Tipo de forma	Comodín		Intervalo de índices	Número de capas (por ejemplo ":L3")	Número de bytes (por ejemplo ":256")
	Sin estado	Basado en sesión			
Forma explícita	Permitido	No permitido	No permitido	Permitido	Permitido
Forma implícita	Permitido	No permitido	Permitido solamente para las sin estado	Permitido	No permitido

C.8.3 Modelo de parte de losa que involucra trenes JPT (tpmodel)

```

tpmodel = "tpmodel" "=" 1#tpmodel-item
tpmodel-item = [codestream-qualifier "," ] tpmodel-element
tpmodel-element = ["-"] tp-descriptor
tp-descriptor = tp-range / tp-number
tp-range = tp-number "-" tp-number
tp-number = tile-number "." part-number
tile-number = UINT
part-number = UINT
    
```

Este campo se puede utilizar para indicar partes específicas de losa que el cliente quisiera añadir o sustraer del modelo caché del servidor. Como en el caso del campo "modelo", se puede utilizar tanto en peticiones basadas en sesión como en las sin estado. En este último caso, el modelo caché está vacío al inicio de la petición y no sobrevive entre peticiones, aunque sigue proporcionando un mecanismo útil para identificar los elementos de imagen que ya están en la caché de cliente.

Cuando un descriptor de parte losa venga precedido por un carácter "-" se dice que es de sustracción, de lo contrario es de adición. Un descriptor de parte losa de adición indica que el cliente ya tiene en su caché la parte losa o el intervalo de

partes losas indicados; el servidor puede añadir estos elementos a su modelo caché. Un descriptor de parte losa de sustracción indica que el cliente no tiene en su caché la parte losa o el intervalo de partes losa indicados; el servidor deberá suprimir estos elementos de su modelo caché.

El primer valor en el número de parte losa es el índice de losa (empezando desde el número 0); el segundo es el número de parte (empezando desde 0) dentro de la losa. Se considera que un tp-range contiene independientemente losas desde el primer número de losa hasta el segundo número de losa y partes de losa del primer número de parte losa hasta el segundo número de parte losa. Así, por ejemplo, 4.0-5.1 incluye las partes losa 4.0, 4.1, 5.0 y 5.1, más no 4.2 ó 5.2.

Tanto el campo de petición "tpmodel" como el "model" puede aparecer dentro de una sola petición. En este caso, no obstante, el servidor deberá reflejar el efecto del campo "model" en su modelo caché antes de procesar el campo "tpmodel".

Se pueden intercalar entre la lista de tpmodel-elements valores de calificador de tren de código con el fin de alterar la recolección de los trenes de código a los cuales se aplican los tpmodel-elements que vienen después, siguiendo exactamente el mismo principio que para el campo de petición "model".

NOTA – Contrario a lo que ocurre con el campo de petición "model", se permiten tanto los intervalos de parte losa como de trenes de código (en los calificadores de tren de código), dentro del campo de petición "tpmodel", sin importar si se trata de una petición basada en sesión o sin estado.

EJEMPLO 1: "tpmodel=4.0,4.1,5.0-6.2" indica que el cliente ya tiene las primeras dos partes losa de losa 4, y las primeras tres partes losa de las losas 5 y 6 en su caché.

EJEMPLO 2: "tpmodel=-4.0-6.254" indica que el cliente no tiene en su caché partes losa de las losas 4, 5 ó 6.

EJEMPLO 3: "tpmodel=3.0,[131-133],4.0,[100],-0.0-65534.254" indica que el cliente tiene parte losa 0 de la losa 3 del tren de código 0 al que se hace referencia en la petición, además de parte losa 0 de la losa 4 de cada uno de los trenes de código 131 a 133, ambos incluidos, y que está borrando todas las partes losa de su caché de tren de código 100.

C.8.4 Necesidad de peticiones sin estado (*need*)

```
need = "need" "=" 1#need-item
```

```
need-item = [codestream-qualifier "," ] bin-descriptor
```

Este campo sólo puede aparecer en peticiones sin estado, es decir aquellas que no incluyan un campo de petición Channel ID. Tiene la misma sintaxis que el campo de petición modelo, salvo que los descriptores de bins no van precedidos por un símbolo "-". El campo de petición "need" no aparecerá dentro de la misma petición que un campo de petición "model" o "tpmodel".

El campo de petición "need" indica el conjunto de bins de datos que representan un interés potencial para el cliente (o los bins de datos que le bastan). No es necesario que el servidor envíe información que no puede ser interesante. Independientemente del volumen que tenga el conjunto de los bins de datos de interés, el servidor enviará solamente la información pertinente a los campos de petición de la ventana seleccionada o de metadatos.

Es posible explicar el efecto del campo "need" en la petición de servidor utilizando el concepto de un modelo temporal de caché. El modelo temporal de caché se inicializa (vacío) inmediatamente antes de que se tramite la petición y se descarta tras la generación de la respuesta. Cuando aparezca un campo "need" en la petición, se añaden todos los bins de datos posibles al modelo caché, tras lo cual todos los elementos a los que se hace referencia mediante los descriptores de bin en el campo "need" se suprimen del modelo caché. Luego, el servidor procesa la ventana seleccionada solicitada, utilizando este modelo caché para establecer cuáles elementos no es necesario enviar al cliente.

Se pueden intercalar calificadores de tren de código entre la lista de los descriptores de bin con el fin de cambiar la recolección de los trenes de código a los cuales se aplica los descriptores de bin que vienen más adelante, siguiendo exactamente el mismo principio que para los campos de petición "model" y "tpmodel".

EJEMPLO 1: "need=M1,H0:20,P0" significa que el cliente necesita todo el bin de metadatos 1, datos del byte 20 del bin de datos 0 de encabezamiento de losa y todo el bin de datos 0 de recinto.

EJEMPLO 2: "need=P1:256,P5:L2" significa que el cliente necesita los datos que van después del byte 256 (o a partir de él) del bin de datos 1 de recinto, y las capas que van más allá de la segunda capa del bin de datos 5 del recinto.

EJEMPLO 3: "need=H*,P*:L3" significa que el cliente necesita todos los bins de datos de encabezamiento de losas que son relevantes para la ventana seleccionada y las capas que van después de la tercera capa de todos los bins de datos de recinto que son pertinentes para la ventana seleccionada.

EJEMPLO 4: "need=t10r0,t*r1:L4" significa que el cliente necesita todas las capas del índice 10 de losa a un nivel de resolución 0, y las capas más allá de la cuarta capa de todas las losas pertinentes a la ventana seleccionada con un nivel de resolución 1.

ISO/CEI 15444-9:2005 (S)

EJEMPLO 5: "need=t*r0-2:L4" significa que el cliente necesita todas las capas desde la capa 4 de todos los bins de datos de recinto en los niveles de resolución de 0 a 2 (0, 1 y 2) en todas las losas y componentes pertinentes para la petición de ventana seleccionada.

EJEMPLO 6: "need=[120-131],r0,[140;143-145],r0-1" significa que el cliente necesita el nivel 0 de resolución en los trenes de código 120 a 131, ambos incluidos, y los niveles de resolución 0 y 1 de los trenes de código 140 y desde 143 hasta 145, inclusive.

C.8.5 Necesidad de parte losa en peticiones sin estado (*tpneed*)

```
tpneed = "tpneed" "=" 1#tpneed-element
tpneed-element = [codestream-qualifier "," ] tp-descriptor
```

Este campo puede estar presente solamente en las peticiones sin estado, es decir aquellas que no incluyen un campo de petición Channel ID. Tiene la misma sintaxis que el campo de petición *tpmodel*, salvo que los *tp-descriptores* no van precedidos por un símbolo "-". El campo de petición "tpneed" no aparecerá dentro de la misma petición que un campo de petición "model" o "tpmodel".

El campo de petición "tpneed" indica el conjunto de partes losa que pueden ser interesantes para el cliente. No es necesario que el servidor envíe información que no tiene un interés potencial. Sin importar el tamaño del conjunto de partes losa que pueden ser interesantes, el servidor deberá enviar solamente información que sea relevante para los campos de petición de ventana seleccionada o el campo de petición de metadatos.

Se puede explicar el efecto del campo "tpneed" en la petición de servidor mediante el concepto de un modelo de caché temporal. Se inicializa el modelo de caché temporal (vacío) inmediatamente antes de tramitar la petición y se descarta tras haber generado la respuesta. Si aparece un campo "tpneed" en la petición, se añaden todas las partes losa y bins de datos posibles en el modelo caché, tras lo cual se suprimen del modelo caché todos los elementos a los que se haga referencia mediante los descriptores de bins en el campo "need" y todas las partes losa en el campo "tpneed". Luego, el servidor tramita la ventana seleccionada solicitada, utilizando este modelo caché para establecer cuáles son los elementos que no es necesario enviar al cliente.

Se pueden intercalar identificadores de tren de código entre la lista de partes losa con el fin de modificar la recolección de los trenes de código a los cuales se aplican las partes losa que vienen a continuación, siguiendo exactamente el mismo principio que para los campos de petición "model" y "tpmodel".

C.8.6 Conjunto de modelos para las peticiones en una sesión (*mset*)

```
mset = "mset" "=" 1#sampled-range
```

Este campo tiene dos propósitos. En primer lugar, sirve para informar al servidor el conjunto de trenes de código para los cuales el cliente está listo para almacenar en la caché los datos administrados por el servidor. Por otra parte, este campo suministra un mecanismo para que el cliente conozca los trenes de código para los cuales el servidor está listo a modelar la caché de cliente. En particular, cuando el conjunto de índices de trenes de código que se suministra en una petición "mset" tiene cualquier diferencia con el conjunto de trenes de código para los cuales el servidor está actualmente preparado a establecer un modelo caché, el servidor deberá emitir un encabezamiento de respuesta Model Set, como se discute en D.2.18.

La cadena de parámetros del campo de petición "mset" está compuesta por una lista separada por comas de intervalos de índices de tren de código, que tal vez hayan sido submuestreadas, siguiendo los convenios que se describen en C.4.6 respecto al campo de petición de tren de código.

Además de los trenes de código que se mencionan en la petición "mset", el servidor también puede proporcionar un modelo caché para todos los trenes de código asociados con su respuesta a la petición en curso. Ésta es la colección de trenes de código identificada por la petición de cliente (véase en C.4.7 los campos de petición de tren de código y contexto de tren de código), a menos que el servidor indique un conjunto reducido de trenes de código mediante un encabezamiento de respuesta de tren de código (véase D.2.9). Cuando no se proporciona el campo de petición "mset", el cliente no podrá suponer que el servidor está proporcionando un modelo caché para todo tren de código que no sea uno de aquellos asociados con su respuesta; sin embargo, es posible modelar otros trenes de código. Si existe un campo de petición "mset", el servidor habrá de descartar toda información de modelo de caché que tenga para los trenes de código que no sean aquellos mencionados bien en la petición "mset" o en el conjunto de trenes de código asociado con sus datos de respuesta. Más aún, el efecto de cualquier manipulación de modelo de caché a través de los campos de petición "model" o "tpmodel" se restringirá estrictamente a dichos trenes de código.

El servidor puede, si así lo desea, reducir el número de trenes de código en el "mset", en cuyo caso habrá de suministrar un encabezamiento de respuesta "mset" que identifique el conjunto real de trenes de código que se está modelando; además, este conjunto de trenes de código que se está modelando tendrá que incluir por lo menos todos los trenes de código asociados con los datos de respuesta del servidor (aquellos solicitados en la petición de cliente o identificados

por el encabezamiento de respuesta de tren de código de servidor, de haberlo). En este caso, estas declaraciones tienen que ver con los trenes de código contenidos en el "mset" identificado por el servidor. Puede ocurrir que el servidor no identifique un conjunto mayor de trenes de código que aquellos mencionados en la petición "mset" de cliente combinados con los asociados con los datos de respuesta de servidor.

Cabe observar que es posible que el servidor cambie su "mset" entre una petición y otra, de tal manera que el cliente que necesite mantener un registro de los "mset" de servidor y/o restringirlos fuertemente deba optar por incluir un campo de petición "mset" en cada petición.

C.9 Parámetros de petición de carga

C.9.1 Carga (*upload*)

```
upload = "upload" "=" upload-type
upload-type = image-return-type ; C.7.3
```

Este campo especifica que el cliente está cargando nuevas imágenes o metadatos del servidor. `upload-type` puede tener cualquier valor entre los valores aceptados de `image-return-type` que se pueden utilizar con el campo de petición de tipo. En el anexo E se da más información acerca de la carga de datos.

C.10 Capacidad de cliente y campos de petición de preferencia

C.10.1 Capacidad de cliente (*cap*)

```
cap = "cap" "=" 1#capability-group
capability-group = processing-capability
                  / depth-capability
                  / config-capability
processing-capability = compatibility-capability
                      / vendor-capability
compatibility-capability = "cc." compatibility-code
vendor-capability = "vc." vendor-code [":" vendor-value]
vendor-code = 1*(LOWER / DIGIT / "." / "-")
vendor-value = TOKEN
depth-capability = "depth:" UINT
config-capability = "config:" UINT
```

Este campo especifica las capacidades del cliente. Cuando se trate de peticiones basadas en sesión (aquellas para las cuales se incluye un campo de petición Channel ID) todo campo de capacidad transmitido por el cliente afectará solamente el canal asociado con la petición y se considerará persistente. Para las peticiones subsiguientes en el mismo canal no es necesario que el cliente retransmita las capacidades.

Cuando se crea un nuevo canal a partir de un canal existente, aquél hereda las capacidades de éste. Cuando se trata de peticiones sin estado, y para las peticiones emitidas dentro de un canal cuyas capacidades nunca han sido especificadas o heredadas, se pueden establecer o determinar las capacidades del cliente utilizando otros medios. Las capacidades asociadas con un canal pueden cambiar al incluir un campo de petición de capacidades de cliente en una petición.

Si el campo de petición capacidades de cliente identifica una o varias de las opciones `processing-capability`, el servidor deberá suponer que el cliente no tiene ninguna de las otras opciones `processing-capability` que puedan haber sido mencionadas. Si no se suministra las opciones `processing-capability` en el campo de petición capacidad del cliente, el servidor deberá seguir utilizando todo conocimiento anterior que tenga al respecto de estas capacidades. Las opciones `processing-capability` definidas por esta Recomendación | Norma Internacional se describen en el cuadro C.6.

Cuadro C.6 – Capacidades válidas del elemento *processing-capabilities*

Capacidad	Significado
<i>compatibility-capability</i>	El cliente soporta todos los ficheros que contienen <i>compatibility-code</i> en la lista de compatibilidades en la casilla del tipo de fichero. Por ejemplo, para indicar que el cliente soporta todos los ficheros JP2, ha de transmitir "cc.jp2_" en un campo de petición de capacidades. Se utilizará un valor de <i>compatibility-code</i> igual a "jp2c" para indicar que se soportan los trenes brutos de código JPEG 2000.
<i>vendor-capability</i>	El cliente soporta las capacidades de fabricante definidas mediante <i>vendor-code</i> , que debe ser una cadena que especifique el nombre de dominio a la inversa del fabricante que define la característica, seguido por el nombre de la característica de vendedor. Por ejemplo, si example.com ha definido una característica llamada "distance", el valor del <i>vendor-code</i> para esta característica será "com.example.distance". <i>vendor-value</i> especifica un valor facultativo, tal como se define mediante la característica de fabricante correspondiente.

Si se suministra un parámetro *depth-capability*, se está indicando la máxima profundidad de bits de muestra (precisión) a la cual el cliente puede utilizar las imágenes sin comprimir. Cuando el cliente soporte diversas profundidades de bit para diferentes componentes de imagen, se ha de especificar en este campo la profundidad de bit de la componente para la cual el cliente posee la mayor capacidad de profundidad de bit.

NOTA 1 – Si un cliente soporta 12 bits para la luminancia y 8 bits para la crominancia, el valor de la capacidad de la profundidad será 12.

NOTA 2 – Los clientes que tengan la capacidad de tratar solamente *N* bits por muestra deberán ser en general, capaces de tratar trenes de código cuyo marcador SIZ indique una profundidad de bit mucho mayor que *N*. Sin embargo, el servidor puede utilizar esta bandera para establecer un mecanismo apropiado que sirva para entregar los datos de imagen solicitada.

Si se suministra un parámetro *config-capability*, éste debe ir en la gama de valores entre 0 y 255, lo que representa una palabra de 8 bits cuyos bits individuales se interpretan como banderas de configuración. En el cuadro C.7 se presenta la interpretación de las banderas de configuración.

Cuadro C.7 – Valores válidos de los parámetros *config-capability*

Valor	Significado
1xxx yyyy	El cliente puede procesar datos de imagen de color
0xxx yyyy	El cliente no puede procesar datos de imagen de color y desea que el servidor transmita toda región de imagen solicitada a escala de grises
x1xx yyyy	El cliente posee un mecanismo de puntería para la interacción con el usuario final
x0xx yyyy	El cliente no tiene un mecanismo de puntería para la interacción con el usuario final
xx1x yyyy	El cliente tiene un teclado para la interacción con el usuario final
xx0x yyyy	El cliente no tiene un teclado para la interacción con el usuario final
xxx1 yyyy	El cliente tiene capacidades de salida de sonido
xxx0 yyyy	El cliente no tiene capacidades de salida de sonido
Otros valores	Reservado para la ISO

Un valor de bit "x" en el cuadro C.7 indica que el valor especificado incluye casos en los que el bit está puesto a "1" o a "0". Los bits que se indican mediante "y" no son utilizados en esta Recomendación | Norma Internacional y los clientes han de ponerlos a 0 mientras que los servidores los deben ignorar.

C.10.2 Preferencias de cliente (*pref*)

C.10.2.1 Generalidades

```

pref = "pref" "=" 1#(related-pref-set ["/r"])

related-pref-set = view-window-pref           ; C.10.2.2
                  / colour-meth-pref         ; C.10.2.3
                  / max-bandwidth           ; C.10.2.4
                  / bandwidth-slice         ; C.10.2.5
                  / placeholder-pref        ; C.10.2.6
                  / codestream-seq-pref     ; C.10.2.7
                  / other

other = TOKEN

```

Este campo indica las preferencias del cliente respecto al comportamiento del servidor. Para las peticiones basadas en sesiones (aquellas que incluyen un campo de petición Channel ID) todo campo de preferencia transmitido por el cliente afectará solamente el canal asociado con la petición y deberá ser considerado persistente. Para las peticiones subsiguientes en el mismo canal, el cliente no necesita retransmitir las preferencias. Cada preferencia aparecerá solamente una vez en un solo campo de petición de preferencia.

Cuando se crea un nuevo canal a partir de un canal existente, se heredan sus preferencias. Para las peticiones sin estado, y para las peticiones emitidas dentro de un canal cuyas preferencias no hayan sido especificadas o heredadas, se pueden establecer las preferencias de cliente o anticiparlas mediante otros métodos. Si el cliente desea cambiar sus preferencias, tendrá que enviar de nuevo el `related-pref-set` en cuestión.

A menos que se señale lo contrario, cada `related-pref-set` especifica una lista ordenada de preferencia mayor a preferencia menor de testigos de preferencia individuales. Siempre que sea posible, el servidor habrá de respetar las preferencias del cliente identificadas por este campo de petición. Si un `related-pref-set` viene seguido por el modificador `"/r"` (requerido), el servidor tendrá que soportar bien una de las preferencias enumeradas en la `related-pref-set` o bien responder con un error. En este último caso, el servidor devolverá un encabezamiento de respuesta de preferencia no disponible que identifica todo `related-pref-set` que tenga el modificador `"/r"` pero no pueda ser soportado. En D.2.20 se dan más detalles acerca del encabezamiento de respuesta de preferencia no disponible.

Por ejemplo, considérese la siguiente petición de preferencias del cliente:

```
pref=fullwindow/r,color-ricc:2;color-icc
```

Esta preferencia hace necesario que el servidor devuelva toda la ventana seleccionada solicitada, independientemente de su longitud (véase C.10.2.2 para una discusión acerca de la preferencia "fullwindow"). Puesto que se ha utilizado el modificador `"/r"`, el servidor debe devolver una respuesta de error a menos que sea capaz de soportarla. Además, el cliente prefiere utilizar perfiles ICC Restricted en lugar de perfiles ICC arbitrarios, siempre que el perfil Restricted ICC sea por lo menos de "calidad excepcional". Véase C.10.2.3 para más detalles acerca de las preferencias de espacio de color.

Un servidor ignorará todo valor de `related-pref-set` que no entienda y que no venga seguido inmediatamente por `"/r"`. Cuando el valor que no se ha entendido venga seguido de `"/r"`, el servidor devolverá el encabezamiento de respuesta de preferencia no disponible, indicando la preferencia que no puede ejecutar.

Se reservan para el uso de la ISO los valores del testigo `other`.

C.10.2.2 Preferencias para el tratamiento de ventana seleccionada

```
view-window-pref = "fullwindow" / "progressive"
```

Esta Recomendación | Norma Internacional define dos opciones para indicar el comportamiento del servidor cuando no pueda atender la petición como se le solicita, siguiendo un orden progresivo en calidad de los datos de respuesta. En el cuadro C.8 se especifican estas dos opciones.

Cuadro C.8 – Preferencias para el manejo de ventanas seleccionadas

Opción	Significado
"fullwindow"	El servidor habrá de satisfacer los parámetros de petición de ventana seleccionada aunque se le permite devolver los datos en un orden que no sea progresivo en la calidad.
"progressive"	El servidor puede modificar los parámetros de petición de ventana seleccionada con el fin de retener las propiedades progresivas en la calidad de los datos de respuesta. Cuando el servidor modifique los parámetros de petición de ventana seleccionada, la ventana seleccionada modificada será un subconjunto de la ventana seleccionada solicitada originalmente.

Cuando no se especifique en las preferencias de cliente ni "ventana completa" ni "progresiva", el servidor deberá suponer que la preferencia del cliente es "progresiva".

Cabe observar que la entrega "progresiva" puede verse afectada por la presencia de un campo de petición de velocidad de entrega, como se explica en C.7.4.

C.10.2.3 Preferencia de método de espacio de color

```
color-meth-pref = 1$(color-meth [":" meth-limit])
```

```
color-meth = "color-enum" / "color-ricc" / "color-icc" / "color-vend"
```

```
meth-limit = UINT
```

Esta Recomendación | Norma Internacional define cuatro opciones que especifican qué formas de datos de especificación de color de espacio han de ser devueltas por el servidor. Un solo fichero JPEG 2000 puede contener múltiples especificaciones del espacio de color para un solo tren de código o una sola capa de composición. Siendo así, un escritor de fichero puede suministrar una especificación óptima de color de espacio sin dejar de tener soluciones que interfueren.

No obstante, no todos los lectores soportarán todos los métodos de espacio de color y los datos proporcionados por alguno de estos métodos pueden ser de un tamaño significativo, en cuyo caso el servidor enviará solamente los datos de especificación de espacio de color deseados por el cliente.

Si el campo de petición preferencias de cliente no contiene ninguna preferencia de método de espacio de color, los métodos de espacio de color soportados se definen conforme a la información incluida en el campo capacidad, y no se define ninguna preferencia.

Cada preferencia de método de espacio de color está compuesta por dos partes, a saber, el método espacio de color propiamente dicho y un límite opcional en dicha preferencia. En el cuadro C.9 se especifican los valores admitidos de método espacio de color.

Cuadro C.9 – Preferencias de cliente de método de espacio de color

Método	Significado
"color-enum"	El cliente prefiere las especificaciones de espacio de color que utilizan el método enumerado (<i>Enumerated Method</i>)
"color-ricc"	El cliente prefiere las especificaciones de espacio de color que utilizan el método ICC restringido (<i>Restricted ICC</i>)
"color-icc"	El cliente prefiere las especificaciones de espacio de color que utilizan el método cualquier ICC (<i>Any ICC</i>)
"color-vend"	El cliente prefiere las especificaciones de espacio de color que utilizan el método del fabricante (<i>Vendor</i>)

El valor opcional `meth-limit` especifica un límite para el valor APPROX en el método de espacio de color que sea del caso. Cuando se utilicen estas preferencias para escoger una especificación de espacio de color, el servidor deberá tener en cuenta una especificación con método de espacio de color cuyo valor APPROX sea `meth-limit` o menor cuando el valor APPROX real sea 1 (exacto). De esta manera, un cliente puede especificar el punto en el cual no es importante la fidelidad de color para determinado método de espacio de color, en la aplicación en cuestión. Por ejemplo, en una aplicación de presentación de página en la que sólo importe la alineación de los datos de imagen con otros elementos de la página puede no ser importante la fidelidad de color y se fija entonces `meth-limit` a 4, es decir que la precisión de los métodos de espacio de color no es importante aquí. Otra aplicación que presente imágenes en una pantalla de baja calidad puede fijar `meth-limit` a 3, para indicar que siempre que la precisión de color sea razonable las condiciones serán satisfechas. Los caracteres del campo se interpretarán como enteros decimales sin signo. Los valores válidos para la definición del campo APPROX se presentan en el cuadro M.24 de la Rec. UIT-T T.801 | ISO/CEI 15444-2, y en las extensiones y enmiendas a dicha Recomendación | Norma Internacional .

Cuando se trata de escoger cuál casilla de especificación de espacio de color se ha de transmitir al cliente, el servidor deberá utilizar el algoritmo mostrado en la figura C.3 a continuación.

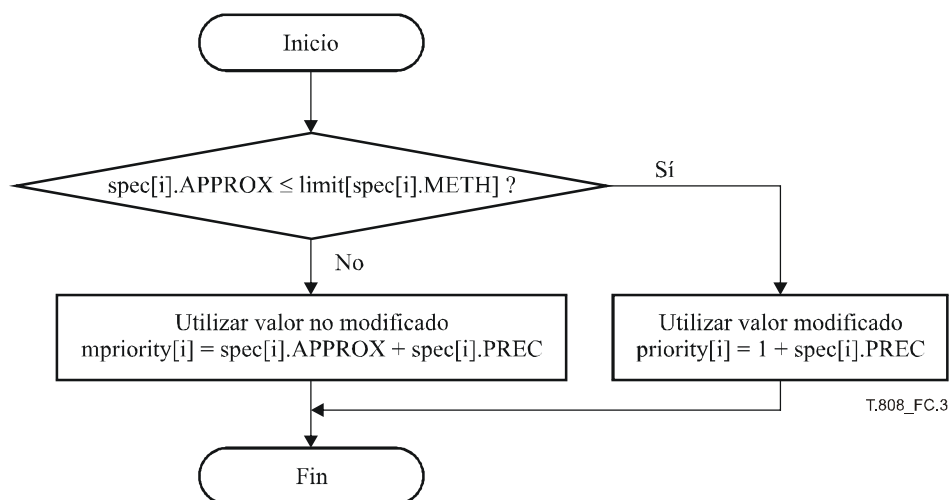


Figura C.3 – Procedimiento de selección de casilla de especificación de espacio de color

Para cada casilla de especificación de espacio de color que utilice un método soportado por el cliente, y donde:

- `spec[]` es una matriz que contiene todas las casillas de especificación de espacio de color de un determinado objetivo lógico.
- `spec[i].APPROX` es el valor del campo APPROX en la casilla de especificación de espacio de color, tal como figura en el objetivo lógico.
- `spec[i].METH` es el valor del campo METH para la *i-ésima* casilla de especificación de espacio de color, tal como figura en el objetivo lógico.
- `spec[i].PREC` es el valor del campo PREC para la *i-ésima* casilla de especificación de espacio de color, tal como figura en el objetivo lógico.
- `limit[]` es una matriz que contiene los valores `meth-limit` especificados en el campo de petición, indexada mediante los valores aceptados del campo METH en la casilla de especificación de espacio de color.
- `priority[]` es una matriz de valores calculados de prioridad para casilla de especificación de espacio de color en determinado objetivo lógico. `priority[i]` corresponde a `spec[i]`.

Si el servidor sabe que el cliente no soporta una determinada casilla de especificación de espacio de color, tendrá que ignorar dicha casilla a efectos de escoger la casilla de espacio de color preferida. Una vez se hayan calculado los valores de `priority[]` para cada casilla de especificación de espacio de color soportada, el servidor deberá escoger la casilla con el menor valor de prioridad. Cuando varias casillas tengan el mismo valor mínimo de prioridad para el objetivo lógico en cuestión, el servidor escogerá el método de espacio de color utilizando el siguiente orden de preferencia.

- 1) Método Enumerated.
- 2) Método Vendor.
- 3) Método Restricted ICC.
- 4) Método Any ICC.

Independientemente de las preferencias de cliente para las casillas de especificación de espacio de color, es posible que el servidor retorne más casillas de especificación de espacio de color que la sola casilla de color especificada por ese algoritmo, dependiendo de si un fichero está dividido en bins de metadatos.

C.10.2.4 Ancho de banda máximo

```
max-bandwidth = "mbw:" mbw
mbw = UINT ["K" / "M" / "G" / "T"]
```

Esta preferencia indica la velocidad máxima a la cual el cliente quisiera recibir datos por objetivo lógico. Si el valor de `mbw` termina en "K", el valor tiene unidades de kilobits/segundo, donde 1 kilobit = 1024 bits; si termina en "M", en megabits/segundo, donde 1 megabit = 1024² bits; si termina en "G", en gigabits/segundo, donde 1 gigabit = 1024³ bits; si termina en "T", en terabits/segundo, donde 1 terabit = 1024⁴ bits. De lo contrario, el valor se considera en bits/segundo. El ancho de banda máximo disponible para el servicio JPIP puede ser limitado también por la capacidad del servidor o de la red.

C.10.2.5 Sector de ancho de banda

```
bandwidth-slice = "slice:" slice
slice = NONZERO
```

Se puede utilizar esta preferencia para identificar la fracción del ancho de banda disponible que debe atribuirse a este canal. El valor de `slice` debe ser mayor que 0. La fracción de ancho de banda se obtiene dividiendo cada valor de sector de canal entre la suma de todos los valores de sector de canal. Si no se especifica, el valor del sector de canal por defecto es 1.

Por ejemplo, un valor `slice` bajo puede ser utilizado para solicitar una ventana seleccionada de "fondo", mientras que uno alto serviría para una de "primer plano". Si la sesión contiene canales asociados con diferentes objetivos lógicos, los valores de sector afectarán la proporción del ancho de banda disponible que se atribuye a esos diferentes objetivos (imágenes).

C.10.2.6 Preferencia de marcador de posición

```
placeholder-pref = "meta:" placeholder-branch
placeholder-branch = "incr" / "equiv" / "orig"
```

Se puede utilizar esta preferencia para indicar el tratamiento preferido de las casillas marcador de posición. Siempre que haya casillas marcador de posición dentro de los metadatos en un tren JPP o JPT, habrá por lo menos tres representaciones diferentes del contenido de una casilla, a saber la casilla original, una casilla equivalente de transmisión y un tren de código incremental (señalado a través del índice). En A.3.6 y A.4 se explican estas posibilidades. En esta última subcláusula, se recomienda suponer por defecto que el cliente prefiere recibir el tren de código incremental, de haberlo, y en caso contrario preferiría recibir la casilla equivalente de transmisión, si la hubiere. El cliente puede señalar otra preferencia utilizando el mecanismo descrito en esta cláusula. Los valores válidos de la preferencia de marcador de posición se especifica en el cuadro C.10.

Cuadro C.10 – Preferencia de marcador de posición

Método	Significado
"orig"	El cliente preferiría recibir la casilla original, de haberla. Si no, preferiría recibir una casilla equivalente de transmisión, si la hubiere.
"equiv"	El cliente preferiría recibir una casilla equivalente de transmisión, de haberla. Si no, el cliente preferiría recibir la casilla original, si la hubiere.
"incr"	El cliente preferiría recibir los bins de datos de tren de código incremental, de haberlo. Si no, preferiría recibir la casilla equivalente de transmisión, si la hubiere. Ésta es la política recomendada por defecto.

No se admite proporcionar más de un valor para la preferencia de marcador de posición.

C.10.2.7 Secuenciamiento de tren de código

```
codestream-seq-pref = "codeseq:" codestream-seq-option
codestream-seq-option = "sequential" / "reverse-sequential"
                        / "interleaved"
```

Esta preferencia se puede utilizar para indicar cómo desea el cliente que el servidor le entregue múltiples trenes de códigos que hayan sido solicitados dentro de una sola petición. En el cuadro C.11 se indican los valores válidos para la preferencia de secuenciamiento de tren de código.

Cuadro C.11 – Preferencias de secuenciamiento de tren de código

Método	Significado
"sequential"	El cliente preferiría recibir los múltiples trenes de códigos en un orden secuencial de cuadros (por ejemplo, ofrecer cuadros múltiples en un fichero JPEG 2000 en movimiento en orden secuencial).
"reverse-sequential"	El cliente preferiría recibir los múltiples trenes de códigos en un orden secuencial de cuadros (por ejemplo, ofrecer cuadros múltiples en un fichero JPEG 2000 en movimiento), en el orden inverso.
"interleaved"	El cliente preferiría recibir los múltiples trenes de códigos de una manera intercalada (por ejemplo, capas de composición múltiples intercaladas por el servidor de un fichero JPX).

No se admite suministrar más de un valor para la preferencia de secuenciamiento de tren de código.

C.10.3 Sensibilidad de contraste (csf)

```
csf = "csf" "=" 1#csf-sample-line
csf-sample-line = csf-density [;" csf-angle] ";" 1$sensitivity
csf-density = "density" ":" UFLOAT
csf-angle = "angle" ":" UFLOAT
sensitivity = UFLOAT
```

Este campo se puede utilizar para suministrar información relativa a la sensibilidad de contraste. Aunque esta información pueda representar los efectos tanto de la sensibilidad visual como de la función de transferencia de modulación de un dispositivo de presentación de imágenes, se la describe más fácilmente en términos de una hipotética función de transferencia de modulación. Cuando se reproduce a la velocidad de cuadros que identifica el campo de petición velocidad de cuadro, se supone que las imágenes han de pasar a través de un dispositivo cuya función de transferencia de modulación (MTF, *modulation transfer function*) es $m(\omega_1, \omega_2)$, tras lo cual es observada por un sujeto cuyo sistema humano de visión tiene una función de sensibilidad de contraste perfectamente uniforme. La MTF $m(\omega_1, \omega_2)$ se describe a través de un conjunto de muestras, que se espacian logarítmicamente en la dirección radial a lo

largo de uno o varios ejes orientados. El servidor puede interpolar estas muestras utilizando cualquier método que le convenga, con el fin de recuperar la MTF, que a su vez se puede utilizar para ajustar el orden en el que se comunican los intervalos de bytes de los bins de datos al cliente a través de mensajes de tren JPP o JPT.

Cada *csf-sample-line* representa muestras MTF $m(\omega_1, \omega_2)$ dado $\omega_1 = \pi d^n \cos \psi$, $\omega_2 = \pi d^n \sin \psi$, donde n es el índice de muestra, que empieza desde $n = 0$ para la primera muestra *csf-density* en *csf-sample-line*, ψ es la orientación de la línea de muestras CSF, expresada en grados (que vale 0 por defecto cuando no haya valor *csf-angle*), y d es la densidad de muestreo que no podrá ser mayor que 1,0. El valor ω_1 describe la frecuencia horizontal en radianes, donde $\omega_1 = \pi$ es la frecuencia horizontal de Nyquist. El valor ω_2 describe la frecuencia vertical en radianes, donde $\omega_2 = \pi$ es la frecuencia vertical de Nyquist.

Los valores de muestra MTF tienen significado solamente cuando se relacionan entre ellos, pues no existe una interpretación determinada para sus valores absolutos.

Anexo D

Señalización de respuesta del servidor

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

D.1 Sintaxis de la respuesta

D.1.1 Introducción

El presente anexo describe todos los elementos que puede contener una respuesta JPIP. Cada subcláusula principal describe el código de estado y su correspondiente descripción de la razón, los encabezamientos de respuesta y los valores posibles de dichos encabezamientos, y los datos de respuesta. En general, una respuesta consta de múltiples encabezamientos de respuesta.

D.1.2 Estructura de la respuesta

La respuesta JPIP consta de los siguientes elementos:

- código de estado;
- descripción de la razón;
- encabezamiento de la respuesta jpip;
- datos de respuesta.

Los elementos en la respuesta deben ser acordes con el protocolo de transporte seleccionado. Por ejemplo, en HTTP, el código de estado y la descripción de la razón aparecen en la línea de estado, los encabezamientos de respuesta JPIP aparecen en los encabezamientos de respuesta HTTP y los datos de la respuesta (si hubiere) figuran en el cuerpo de la entidad HTTP.

Status-Code = 3DIGIT

Reason-Phrase = *<TEXT, excluding CR and LF>

```

jpip-response-header =
    / JPIP-tid                ; D.2.2
    / JPIP-cnew               ; D.2.3
    / JPIP-qid                ; D.2.4
    / JPIP-fsiz               ; D.2.5
    / JPIP-rsiz               ; D.2.6
    / JPIP-roff               ; D.2.7
    / JPIP-comps              ; D.2.8
    / JPIP-stream             ; D.2.9
    / JPIP-context            ; D.2.10
    / JPIP-roi                ; D.2.11
    / JPIP-layers             ; D.2.12
    / JPIP-srate              ; D.2.13
    / JPIP-metareq            ; D.2.14
    / JPIP-len                ; D.2.15
    / JPIP-quality            ; D.2.16
    / JPIP-type               ; D.2.17
    / JPIP-mset               ; D.2.18
    / JPIP-cap                ; D.2.19
    / JPIP-pref               ; D.2.20

```

Lo ideal es que la cadena de la descripción de la razón explique en palabras el código de estado. Los siguientes códigos de estado pueden ser suficientes para aplicaciones JPIP.

D.1.3 Códigos de estado y descripciones de razón

D.1.3.1 Consideraciones generales

El Status-Code es un código entero de 3 cifras que resulta del intento de comprender y atender la petición. Se utiliza un subconjunto de los códigos de estado y las descripciones de razón del protocolo HTTP/1.1. Los clientes JPIP deben esperar recibir los códigos que se describen a continuación. No obstante, los clientes JPIP que operen sobre el HTTP pueden recibir además otros códigos de estado.

D.1.3.2 200 (OK)

El servidor utiliza este código de estado si acepta la petición del procesamiento de la ventana seleccionada, posiblemente con ciertas modificaciones de la ventana seleccionada, que se indican en los encabezamientos adicionales incluidos en la respuesta.

D.1.3.3 202 (Aceptado)

Los servidores responden con este código de estado si la petición de la ventana seleccionada era aceptable, pero había en la cola una petición de ventana seleccionada posterior que anulaba la primera (porque `wait=no`). Cuando la primera petición pasa a ser irrelevante antes de que el servidor pueda procesar y comience a transmitir una respuesta, se utilizará el código de estado 202. Esto sucede frecuentemente en la práctica, dado que un usuario interactivo puede cambiar su región de interés varias veces antes de que el servidor termine de responder una petición anterior, o antes de que el servidor esté preparado para interrumpir un procesamiento en curso.

D.1.3.4 400 (Petición incorrecta)

Los servidores deben contestar con este código de estado si el formato de la petición es incorrecto o contiene un campo no reconocido en la cadena de la consulta.

D.1.3.5 404 (No se encuentra)

Se utiliza este código de estado cuando el servidor no puede hacer cuadrar el recurso solicitado con un ID de destino expresado. Esto puede deberse a intentos de acceso no autorizado o, más probablemente, a la expiración de una temporización. Si el cliente pierde esa ventana de tiempo, debido a una conexión lenta, puede suceder que el ID de destino ya no esté activo.

D.1.3.6 415 (Tipo de medio no soportado)

Este código de estado puede utilizarse si el tipo de imagen concreto especificado en el campo de respuesta tipo de retorno de imagen (Image Return Type) no puede ofrecerse.

D.1.3.7 501 (No implementado)

Este código de estado puede utilizarse si una parte de esta Recomendación | Norma Internacional que se solicita en la respuesta no puede ofrecerse.

D.1.3.8 503 (Servicio no disponible)

Este código de estado puede utilizarse si el ID del canal especificado en el campo petición ID del canal no es válido.

D.2 Encabezamiento de respuesta JPIP**D.2.1 Introducción a los encabezamientos de respuesta JPIP**

Al responder a una petición del cliente, el servidor puede modificar ciertos aspectos de la petición. Si el servidor modifica la petición, los parámetros modificados deben indicarse en los encabezamientos de respuesta. El nombre de cada encabezamiento de respuesta se obtiene a partir del nombre del campo de la petición cuyos parámetros se están modificando, añadiéndose el prefijo "JPIP-" al nombre del campo solicitado. A no ser que se especifique lo contrario, si los parámetros indicados en el encabezamiento de respuesta fueron especificados originalmente en la petición del cliente, el servidor tendrá que responder de la misma manera, salvo cuando la respuesta no contenga nuevamente estos encabezamientos de respuesta. Por otra parte, el servidor puede enviar los encabezamientos de respuesta JPIP para informar al cliente de los valores de otros campos de petición no especificados que podrán utilizarse en futuras peticiones.

La respuesta JPIP-`qid` es una excepción, puesto que deberá enviarse siempre que el cliente incluya una ID de petición en la petición, y el valor de JPIP-`qid` sea siempre igual a `qid`.

Los parámetros del encabezamiento de respuesta derivado que se indica como parámetro por el mismo elemento BNF en el campo de petición original tienen el mismo significado y formato que los parámetros del campo de petición original.

Las únicas excepciones a esta regla son las relativas a los encabezamientos de respuesta nuevo canal y calidad.

D.2.2 ID de destino (JPIP-tid)

```
JPIP-tid = "JPIP-tid" ":" LWSP target-id
```

ISO/CEI 15444-9:2005 (S)

El servidor enviará este encabezamiento de respuesta si el identificador de destino unívoca del servidor difiere de alguna manera del identificador facilitado en el campo petición ID de destino o el cliente no especifica el campo el campo petición ID de destino. El `target-id` es una cadena cualquiera asignada por el servidor cuya longitud no excederá de 255 caracteres. Si el campo petición ID de destino especifica un valor de "0", el servidor está obligado a incluir el encabezamiento de respuesta ID de destino, en el cual se indica el `target-id`. Si el servidor no puede asignar identificadores únicos a los destinos lógicos solicitados y por consiguiente no puede garantizar su integridad entre múltiples peticiones o sesiones, el encabezamiento de respuesta del ID de destino especificará un valor de 0. Si el servidor proporciona un `target-id` que es diferente del especificado en la petición, el servidor hará caso omiso de todos los campos de petición `model`, `tparamodel`, `need` y `tparam` cuando responda a dicha petición.

D.2.3 Nuevo canal (JPIP-cnew)

```
JPIP-cnew = "JPIP-cnew" ":" LWSP "cid" "=" channel-id  
          [", " 1#(transport-param "=" TOKEN)]
```

```
transport-param = TOKEN
```

El servidor enviará este encabezamiento de respuesta si, y sólo si, asigna un nuevo canal en la respuesta al campo de petición a un nuevo canal. La cadena de valores consistirá en una lista separada por coma de pares nombre=valor, el primero de los cuales identifica el testigo ID del canal del nuevo canal.

En esta Recomendación | Norma Internacional se definen los siguientes testigos `transport-param` (véase el cuadro D.1).

Cuadro D.1 – Valores válidos de transport-param

Valor	Significado
"transport"	Este parámetro se asignará a uno de los valores en la lista de nombres de transporte aceptables suministrado por el campo petición de nuevo canal. Si se proporcionan múltiples nombres de transporte en el campo petición, el encabezamiento de respuesta identificará el transporte real que habrá de utilizarse dentro del canal.
"host"	Este parámetro identifica el nombre o dirección IP del computador para el servidor JPIP que gestiona el nuevo canal. No es necesario devolver el parámetro a no ser que el computador sea diferente de aquél al que se envió realmente la respuesta.
"path"	Este parámetro indica el componente de trayecto del URL que se utilizará al construir futuras peticiones en este canal. No es necesario devolver este parámetro a no ser que el nombre del trayecto difiera del utilizado en la petición que se envió realmente.
"port"	Este parámetro identifica el número de puerto numérico (decimal) que atiende el servidor JPIP que está gestionando el nuevo canal para leer las peticiones. No es necesario devolver el parámetro si el computador o el número de puerto es idéntico a aquél por el que se envió la petición original. Tampoco es necesario devolver el parámetro si el computador es distinto de aquél por el que se envió la respuesta y se va a utilizar el número de puerto por defecto correspondiente al transporte seleccionado.
"auxport"	Este parámetro se utiliza cuando se requiere un segundo canal físico. Si se emplea el transporte "http-tcp", el puerto auxiliar se utiliza para conectar el canal TCP auxiliar. Para mayor información, véase el anexo G. No es necesario devolver este parámetro si la petición original implica un canal que también utiliza un canal auxiliar, y cuyo número de puerto auxiliar es el mismo. De lo contrario, el parámetro se devolverá únicamente si el número de puerto auxiliar difiere del valor por defecto correspondiente al transporte seleccionado.

D.2.4 ID de petición (JPIP-qid)

```
JPIP-qid = "JPIP-qid" ":" LWSP UINT
```

El servidor enviará este encabezamiento de respuesta si la petición del cliente incluye una ID de petición `qid`. El valor de `JPIP-qid` será idéntico al de `qid`. El servidor no incluirá un encabezamiento de respuesta ID de petición cuando la correspondiente petición de cliente no incluya una ID de respuesta.

NOTA – La ID de respuesta del servidor, `JPIP-qid`, será siempre idéntico al ID de petición del cliente. Por consiguiente, el ID de petición se caracteriza porque este encabezamiento de respuesta se envía cuando el cliente ha utilizado el ID de petición, y no cuando el servidor modifica el valor.

D.2.5 Tamaño del cuadro (JPIP-fsiz)

```
JPIP-fsiz = "JPIP-fsiz" ":" LWSP fx ", " fy
```

El servidor debe enviar este encabezamiento de respuesta si el tamaño del cuadro de los datos de la respuesta del servidor difiere del solicitado en el campo petición del tamaño de cuadro.

D.2.6 Tamaño de región (JPIP-rsiz)

```
JPIP-rsiz = "JPIP-rsiz" ":" LWSP sx "," sy
```

El servidor debe enviar este encabezamiento de respuesta si el tamaño de la región de los datos que responde el servidor difiere del solicitado.

D.2.7 Traslación (JPIP-roff)

```
JPIP-roff = "JPIP-roff" ":" LWSP ox "," oy
```

El servidor debe enviar este encabezamiento de respuesta si la traslación de la región de los datos que responde el servidor difiere de la solicitada.

D.2.8 Componentes (JPIP-comps)

```
JPIP-comps = "JPIP-comps" ":" LWSP 1#UINT-RANGE
```

El servidor debe enviar este encabezamiento de respuesta si los componentes de los datos que envía el servidor difieren de los solicitados en el campo petición de componentes. El servidor no está obligado a enviar este encabezamiento de respuesta si los componentes de la imagen solicitada no existen en ninguno de los trenes de códigos solicitados.

D.2.9 Trenes de código (JPIP-stream)

```
JPIP-stream = "JPIP-stream" ":" LWSP 1#(prefixed-range / sampled-range)
prefixed-range = "<" ctxt-id ":" ctxt-elt ">" sampled-range
ctxt-id = UINT
ctxt-elt = UINT
```

El servidor debe enviar este encabezamiento de respuesta para informar al cliente del tren o trenes de código para los que envía datos, a no ser que el servidor esté enviando datos en respuesta a todos los trenes de código solicitados directamente mediante el campo de petición de trenes de código y todos los trenes de código solicitados indirectamente mediante el campo petición de contexto del tren de código. El servidor debe utilizar la sintaxis `prefixed-range` para indicar los trenes de código para los que está enviando datos en respuesta a un campo petición de contexto del tren de código trasladado. En ese caso, el valor `ctxt-id` indicará el `context-range` específico del campo de petición de contexto del tren de código cuya traslación está produciendo los trenes de código correspondientes. Además, el valor `ctxt-elt` identificará el elemento particular dentro del `context-range` indicado mediante `ctxt-id`, cuya traslación está produciendo los trenes de código correspondientes.

El valor 0 de `ctxt-id` significa que el primer `context-range` en el campo de petición de contexto del tren de código es aquel que produce la gama de trenes de código que siguen al prefijo. Análogamente, el valor de 1 de `ctxt-id` significa que el segundo `context-range` en el campo de petición de contexto de tren de código es aquel que produce las subsiguientes gamas de trenes de código, etc.

El valor de 0 de `ctxt-elt` significa que el primer contexto en el correspondiente `context-range` es aquel que produce la gama de trenes de código que figuran a continuación del prefijo.

Ejemplo:

Petición de cliente:

```
stream=0&context=jpxl<2-7:2>[s0i0],jpxl<3-5>[s1i3]
```

Respuesta del servidor:

```
JPIP-context: jpxl<2-7:2>[s0i0]=0,1;jpxl<9-10>[s1i3]=0
```

```
JPIP-stream: 0,<0:1>1,<1:0>0,<1:1>0
```

Esto significa que el servidor responde con los datos que resultan de:

- 1) la aplicación directa de la ventana seleccionada al tren de código 0 (solicitado por "stream=0");
- 2) la traducción de la ventana seleccionada a la capa 4 de composición JPX, de conformidad con la instrucción de composición 0 en el conjunto de instrucciones de composición 0, y aplicado al tren de código 1;
- 3) la traducción de la ventana seleccionada a la capa 9 de composición JPX, de conformidad con la instrucción de composición 3 en el conjunto de instrucciones de composición 1, y aplicado al tren de código 0; y
- 4) la traducción de la ventana seleccionada a la capa 10 de composición JPX, de conformidad con la instrucción de composición 3 en el conjunto 1 de instrucciones de composición, y aplicado al tren de código 0.

D.2.10 Contexto de tren de código (JPIP-context)

```
JPIP-context = "JPIP-context" ":" LWSP 1$(context-range "=" 1#sampled-range)
```

El servidor debe enviar este encabezamiento de respuesta si es capaz de procesar cualquiera de los valores `context-range` suministrados en el campo de petición de contexto de tren de código. El encabezamiento describe cada `context-range` que se está procesando, junto con los índices de todos los trenes de código que están relacionados con dicho `context-range`. El servidor puede omitir algunos valores de `context-range` que fueron originalmente proporcionados en el campo de petición Contexto de tren de código, si no se están procesando. El servidor también puede modificar los valores de `context-range` proporcionados originalmente en el campo de petición Contexto de tren de código. Son posibles dos tipos de modificación:

- el servidor puede restringir el conjunto de elementos de imagen (por ejemplo, capas de composición) que fueron originalmente solicitadas;
- el servidor puede rechazar modificaciones de transformación geométrica que no es capaz de aplicar (por ejemplo, un modificador "banda de sonido" o "película" dentro de una cadena `mj2t-context`).

D.2.11 ROI (JPIP-roi)

```
JPIP-roi = "JPIP-roi" ":" LWSP
         "roi" "=" region-name ";"
         "fsiz" "=" UINT "," UINT ";"
         "rsiz" "=" UINT "," UINT ";"
         "roff" "=" UINT "," UINT ";"

region-name = 1*(DIGIT / ALPHA / "_")
```

En respuesta a una petición de cliente de una ROI (región de interés), el servidor verificará en el encabezamiento de respuesta ROI las dimensiones de la ROI que en realidad está ofreciendo. Si el servidor es incapaz de cumplir la petición ROI, responderá con el encabezamiento de respuesta ROI fijada al valor: "JPIP-roi: roi=no-roi". Además de la ROI, el servidor también especifica la región de la imagen que está realmente ofreciendo mediante los encabezamientos de respuesta tamaño del cuadro, tamaño de la región y traslación.

Si el servidor es capaz de ofrecer la ROI, pero por alguna razón necesita redimensionar la porción de la imagen que devuelve, enviará el encabezamiento de respuesta ROI que describe la ROI y los encabezamientos de respuesta tamaño del cuadro, tamaño de la región y traslación que describen la parte de la ROI que devuelve.

D.2.12 Capas (JPIP-layers)

```
JPIP-layers = "JPIP-layers" ":" LWSP UINT
```

El servidor debe enviar este encabezamiento de respuesta si el número de capas que está ofreciendo es menor que el valor especificado por el campo de petición capas. Dado que la ventana seleccionada se envía normalmente con calidad progresiva, el servidor no está obligado (y quizás no sea capaz) de determinar el número de capas de que constan los datos de respuesta que está ofreciendo. Ahora bien, si el número solicitado de capas supera el número de capas disponible de cualquier tren de código en la ventana seleccionada, el servidor debería al menos identificar el número máximo de capas disponibles. Todo servidor que acepte el campo petición de Alineación (véase C.7.1) deberá proporcionar una respuesta de capas JPIP si el número de capas que está ofreciendo es menor que el valor especificado por el campo petición de capas.

D.2.13 Velocidad de muestreo (JPIP-srate)

```
JPIP-srate = "JPIP-srate" ":" LWSP UFLOAT
```

El servidor debe enviar este encabezamiento de respuesta si se prevé que la velocidad de muestreo media del tren de código que envía al cliente será diferente de la solicitada en el campo de petición velocidad de muestreo y esta velocidad de muestreo se conoce. Si el tren de códigos de origen no dispone de información de temporización, este encabezamiento de respuesta no deberá enviarse.

D.2.14 Petición de metadatos (JPIP-metareq)

```
JPIP-metareq = "JPIP-metareq" ":" LWSP
              1#( "[" 1$(req-box-prop) "]" [root-bin] [max-depth] )
              [metadata-only]

req-box-prop = box-type [limit] [metareq-qualifier] [priority]
```

El servidor debe enviar este encabezamiento de respuesta si modifica el valor de `max-depth`, `limit`, `metareq-qualifier` o `priority` proporcionados en el campo de petición Petición de metadatos.

D.2.15 Longitud máxima de la respuesta (JPIP-len)

```
JPIP-len = "JPIP-len" ":" LWSP UINT
```

El servidor debe enviar este encabezamiento de respuesta si el límite de bytes especificado en el campo petición de longitud máxima de la respuesta es demasiado pequeño para permitir una respuesta no vacía, a no ser que el límite de bytes fuera igual a cero. En caso de que se devuelva, `JPIP-len` será el valor que informe al cliente de una longitud máxima de la respuesta adecuada, para la subsiguiente respuesta. Si `len=0`, el servidor debe responder a la solicitud con los encabezamientos de respuesta y sin datos de respuesta.

D.2.16 Calidad (JPIP-quality)

```
JPIP-quality = "JPIP-quality" ":" LWSP (1*2DIGIT / "100" / "-1")
```

El servidor debe enviar este encabezamiento de respuesta para informar al cliente del valor de la calidad que tendrán los datos de la imagen que devuelva una vez que la petición se haya completado. Si la petición es interrumpida por otra petición (que no tiene "wait=yes"), este valor de la calidad quizá no sea preciso. El valor de la calidad se refiere únicamente a la ventana seleccionada, y tiene la misma interpretación que el campo petición de calidad. Si el servidor ignora la petición del cliente, se devolverá el valor "-1".

D.2.17 Tipo de retorno de imagen (JPIP-type)

```
JPIP-type = "JPIP-type" ":" LWSP image-return-type
```

El servidor debe enviar este encabezamiento de respuesta a no ser el subtipo MIME de los datos de la imagen devueltos se identifique mediante otro mecanismo. Ejemplos de otros mecanismos son:

- un encabezamiento "Content-Type:" del HTTP;
- las respuestas a las peticiones que están relacionadas con una sesión cuyo tipo de imagen de retorno ya se ha indicado.

D.2.18 Conjunto de modelos (JPIP-mset)

```
JPIP-mset = "JPIP-mset" ":" LWSP 1#sampled-range
```

El servidor debe enviar este encabezamiento de respuesta si la petición del cliente contiene el campo petición de Conjunto de modelos, y el conjunto de trenes de código identificado por el campo petición de conjunto de modelos del cliente difiere de alguna manera del conjunto de trenes de código para el cual el servidor está realmente preparado para mantener la información del modelo de caché. El conjunto de trenes de código para el cual el servidor mantiene la información del modelo de caché debe incluirse en todos los trenes de código que están relacionados con los datos de respuesta del servidor (ya se trate de los identificados en la petición del cliente o en el encabezamiento de respuesta del tren de código de servidor, si la hubiera). Además de estos trenes de código, el "mset" del servidor quizá no sea mayor que el identificado por el campo petición de Conjunto de modelos del cliente.

D.2.19 Capacidad necesaria (JPIP-cap)

```
JPIP-cap = "JPIP-cap" ":" LWSP 1#capability-code
```

Este encabezamiento de respuesta especifica que el cliente deberá soportar una determinada característica a fin de interpretar de manera conforme el destino lógico. Las capacidades autorizadas son las mismas que las definidas en el campo de petición capacidad del cuadro C.6.

D.2.20 Preferencias no disponible (JPIP-pref)

```
JPIP-pref = "JPIP-pref" ":" LWSP 1#related-pref-set
```

Este encabezamiento de respuesta debe proporcionarse si, y sólo si, el campo de petición de preferencias del cliente contiene un `related-pref-set` con el modificador "r" (obligatorio), que el servidor fue incapaz de soportar. En este caso, debe devolverse también el valor del error para el código de estado de la respuesta. La cadena de valores consta de uno o varios `related-pref-sets` que no pudieron soportarse, repetidos de la misma manera que aparecen en la petición de preferencias del cliente.

Si bien es conveniente, no es necesario que el encabezamiento de respuesta enumere todos los `related-pref-sets` requeridos que no puedan soportarse. Por consiguiente, se permite que el servidor se desplace por el campo de petición de personalización del cliente mientras no encuentre un `related-pref-set` con el modificador "r" y no pueda soportarse. Para mayor información sobre cuándo debe utilizarse este encabezamiento de respuesta, véase C.10.2.1.

Cuadro D.2 – Códigos de razón definidos

Código de razón	Razón	Explicación
1	Imagen terminada	El servidor ha transferido al cliente toda la información disponible de la imagen (y no sólo la información correspondiente a la ventana seleccionada). Este código de razón tiene un significado especial para las peticiones basadas en sesión, a saber, este código de razón implica que el cliente ha recibido todos los datos que pueden enviarse en respuesta a cualquier petición basada en sesión relacionada con este destino lógico. Con la única excepción de las peticiones que incluyen campos de peticiones de gestión de caché, se constatará a toda subsiguiente petición basada en sesión sin incluir datos de respuesta y con R=1 EOR.
2	Ventana concluida	El servidor ha transferido toda la información disponible relativa a la ventana seleccionada. Este código de razón tiene un significado especial para las peticiones basadas en sesión, a saber, este código de razón implica que el cliente ha recibido todos los datos que podrían enviarse en respuesta a esta petición y los datos de respuesta no estaban limitados por campo de límite de datos alguno (<i>Length or Quality</i>) en la petición o mediante la gestión de una subsiguiente petición. Salvo para el caso de las peticiones que incluyen campos de petición de gestión de caché, se responderá a las subsiguientes repeticiones de la petición sin incluir datos de respuesta y con R=2 EOR.
3	Cambio de ventana	El servidor está concluyendo su respuesta para atender a una nueva petición que no especifica <i>Wait=yes</i> .
4	Límite de byte alcanzado	El servidor está terminando su respuesta porque el límite de byte especificado en el campo de petición longitud máxima de la respuesta ha sido alcanzado.
5	Límite de calidad alcanzado	El servidor está terminando su respuesta porque se ha alcanzado el límite de calidad especificado en el campo de petición calidad.
6	Límite de sesión no alcanzado	El servidor está terminando su respuesta porque se ha alcanzado algún límite de los recursos de sesión, por ejemplo una temporización. No debe hacerse ninguna petición utilizando el ID de canal correspondiente a dicha sesión.
7	Límite de respuesta alcanzado	El servidor está terminando su respuesta porque se ha alcanzado algún límite, por ejemplo, temporización. Si la petición se hace en una sesión, las siguientes peticiones todavía pueden hacerse utilizando el ID del canal correspondiente a dicha sesión.
0xFF	Razón no especificada	El servidor está terminando su respuesta por una razón que no se especifica.
Otros valores		Reservada para uso ISO

D.3 Datos de respuesta

Para los tipos de retorno de imagen de tren distinto de JPP o JPT, incluidos los trenes de código en bruto, los datos de respuesta deben consistir en la entidad solicitada en su totalidad. Para los tipos de retorno de imagen de tren JPP o JPT, los datos de respuesta consistirán en una secuencia de mensajes definidos en el anexo A, que termina por un único mensaje EOR (Fin de respuesta). El mensaje EOR no se define en el anexo A y formalmente no forma parte de los tipos de medios de tren JPP o JPT.

El mensaje EOR consta de un encabezamiento y un cuerpo. El encabezamiento del mensaje EOR consta de un único identificador de byte, 0x00, seguido de un sólo byte de código de razón, R, y de un sólo contador de bytes VBAS, que indica el número de bytes en el cuerpo del mensaje EOR. En esta Recomendación | Norma Internacional no se proporciona una interpretación normativa del contenido del cuerpo del mensaje EOR.

Obsérvese que el cuerpo del mensaje EOR no contribuye a la restricción del cómputo de bytes relacionada con el campo petición de longitud máxima de la respuesta, definido en el anexo C.

Obsérvese que el mensaje EOR significa que el servidor ha entregado todo el contenido pertinente de los correspondientes bins de datos para la petición de cliente. Éste no es necesariamente el contenido íntegro de dichos bins de datos. La respuesta termina cuando se alcanza el límite específico de cliente. Si no se especificó límite alguno, el mensaje EOR significa que todo el contenido de los correspondientes bins de datos ha sido transmitido.

Los códigos de razón ya están definidos (véase el cuadro D.2).

Anexo E

Carga de imágenes en el servidor

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

E.1 Introducción

Es de esperar que las imágenes se almacenarán en un servidor de diversas maneras, lo cual queda fuera del alcance de esta Recomendación | Norma Internacional. El objetivo de este anexo es describir los mecanismos que permiten cargar en un servidor partes de una imagen.

E.2 Petición de carga

E.2.1 Estructura de la petición

Una petición de carga consiste en uno o varios campos de petición definidos en el anexo C, y un cuerpo de petición.

E.2.2 Campos de la petición de carga

Los campos de petición para realizar una carga contendrán un campo petición de carga. Asimismo, podrán utilizarse los campos petición de destino, subdestino e ID de destino (véanse C.2.2, C.2.3 y C.2.4). Al cargar un tipo de medios de imagen completa, se utilizarán los campos de petición tamaño de cuadro, traslación y tamaño de la región (véanse C.4.2, C.4.3 y C.4.4) para indicar la posición de la porción que se desea cargar dentro de la imagen completa. Al cargar trenes JPT o trenes JPP, son innecesarios el número de bins de datos (y por consiguiente el número de losa o recinto) y con el encabezamiento principal que indica la ubicación de los datos codificados y de los campos de petición de ventana seleccionada.

E.2.3 Cuerpo de la petición de carga

E.2.3.1 Consideraciones generales

El cuerpo de una petición de carga consiste en uno de los tipos de imagen soportados: tren JPP, tren JPT o un tipo de medio de imagen completa. El cuerpo contiene los datos que el cliente solicita que el servidor le gestione. Esta Recomendación | Norma Internacional no soporta la carga de datos de imágenes en bruto.

E.2.3.2 Tren JPT

El cuerpo de la petición contiene todos los bins de datos que el cliente desea reemplazar en el servidor (bins de datos de cabeceras, bins de metadatos y bins de datos de sector). Si el cliente no carga un bin de datos del encabezamiento principal, los bins de datos de sector se codificarán de manera compatible con el encabezamiento principal existente.

E.2.3.3 Tren JPP

El cuerpo de la petición contiene todos los bins de datos que el cliente desea que reemplace el servidor (bins de datos de encabezamientos, bins de metadatos y bins de datos de encabezamiento de sector, bins de metadatos y bins de datos de recinto). Si el cliente no carga un bin de datos del encabezamiento principal o el bin de datos del encabezamiento de segmento, los recintos deben codificarse de manera compatible con los encabezamientos de sector y principal existentes.

E.2.3.4 Carga de una imagen completa

El cuerpo de la petición contiene un tipo de medios de imagen concreta que representa aquellas muestras que el cliente desea modificar.

En el caso de una carga de la imagen completa, la petición puede incluir los campos petición de tamaño del cuadro, región del cuadro y traslación. El campo petición de tamaño del cuadro será el tamaño de la cuadrícula de referencia de la imagen. En el caso de la carga de una imagen completa, la compresión no tiene por qué hacerse de manera compatible con el destino lógico en el servidor. Si el tamaño de la imagen cargada es mayor que las dimensiones del campo petición de tamaño de región, el servidor debe limitar las modificaciones de las dimensiones especificadas en el campo de petición de tamaño de la región.

E.3 Respuesta del servidor

E.3.1 Consideraciones generales

El servidor responderá a una petición de carga con el código de estado y la descripción de la razón descritas en el anexo D. En las subcláusulas que figuran a continuación se describen los códigos de retorno y las descripciones de la razón para la carga de imágenes.

E.3.2 201 (creada)

El servidor utiliza este código de estado si tras recibir una petición de carga, se ha definido un nuevo recurso en el servidor. El servidor ha de haber completado la creación antes de devolver esta petición. Si hay un retraso, el servidor debe devolver 202 (aceptado) en lugar de 201 (creado).

El servidor debe incluir en la respuesta un encabezamiento que contenga el campo del nuevo ID de destino para el recurso modificado.

No es necesario devolver nada en el campo de la respuesta.

E.3.3 202 (aceptado)

El servidor utilizará este código de estado si la carga crea un nuevo recurso pero el servidor no está preparado para ello. El servidor también puede utilizar este código de estado para actualizar un recurso existente.

E.3.4 400 (petición incorrecta)

El servidor debe devolver este código de estado si el formato de la petición es incorrecto o la consulta contiene campos de petición que son incompatibles con la carga, o contiene un campo no reconocido en la cadena de consulta.

E.3.5 404 (desconocido)

El servidor devolverá este código de estado cuando no pueda encontrar la correspondencia de recurso solicitado con el ID de destino recibido.

E.3.6 415 (tipo de medios no soportados)

Este código de estado puede devolverse para indicar que si bien se soportan las cargas, no se soporta, en cambio, el tipo concreto (por ejemplo, imagen completa, tren JPT o tren JPP) incluido en la petición.

E.3.7 501 (No implementado)

Este estado puede utilizarse si el servidor no soporta la carga o no soporta una opción concreta de la carga.

E.4 Fusión de datos en el servidor

E.4.1 Modificación de la imagen

Tras recibir los datos de la carga, el servidor puede crear una nueva versión del destino lógico y proporcionar una nueva versión a los clientes a través de una URL nueva o antigua. Ahora bien, el servidor no debe utilizar el campo de petición de ID de destino antiguo para proporcionar acceso a los datos fusionados o actualizados.

Si el cliente incluye el campo de petición ID de destino en la posición de carga y el ID de carga no corresponde con el ID de carga actual del servidor para ese recurso, el servidor no debe actualizar la imagen. Esta discordancia puede indicar que el cliente ha editado una versión anterior de la imagen que ya ha sido modificada. Los servidores no deberán aceptar las cargas que no contengan un campo de petición de ID de destino. Esto es una manera de prevenir múltiples ediciones simultáneas de un destino por diferentes clientes. Los servidores que ofrecen capacidades de edición pueden controlar estos problemas mediante el bloqueo del destino o por otros mecanismos.

El cliente JPIP puede cargar parte de una nueva imagen mediante la especificación de un ID de destino de valor igual a 0, o utilizar una nueva URL, o bien un destino que el servidor no tiene. El servidor debe asignar un ID de destino para la carga. Un cliente puede continuar cargando porciones adicionales de una nueva imagen utilizando para ello el ID de destino que devuelve el servidor en la carga anterior.

E.4.2 Tren JPT

Un servidor que acepta bins de datos de losa, debe en primer lugar suprimir todos los datos de bins de datos de losa antiguos para esos sectores que se están cargando, y posteriormente incluir los datos del nuevo bin de datos de losa en el tren de código. No puede hacerse una actualización que dé lugar a un cambio en el número, dimensión o ubicación de los sectores: la estructura de la imagen no puede modificarse mediante una carga. En particular, un servidor no debe aceptar cargas de bins de datos de sector para un tren de códigos que contenga el segmento de marcador PPM en el

encabezamiento principal, a no ser que el cliente proporcione un nuevo encabezamiento en la carga. Todos los segmentos de marcador PLM o TLM deben suprimirse o actualizarse. Para las nuevas imágenes deben cargarse los bins de datos de encabezamiento principal JPT.

No se especifican cómo se forma la parte de losa del tren de código a partir de los bins de datos de la losa. El cliente no necesita obligatoriamente proporcionar toda la parte de losa de una losa, ni tampoco completar la última parte de losa. El servidor actualizará el encabezamiento principal y las partes del formato de fichero afectadas (por ejemplo la longitud de la casilla del tren de código).

Al fusionar datos, el número o tamaño de losas no debe cambiar y los datos que no se han sustituido mediante el proceso de carga deberán tener el mismo significado que tenían antes de la carga.

E.4.3 Tren JPP

Cuando un servidor acepte mensajes de bins de datos de recinto deberá, en primer lugar, suprimir los correspondientes bins de datos del antiguo recinto, es decir, los del recinto que se están cargando, y posteriormente incluir los nuevos datos de bins de datos de recinto. No puede modificarse un encabezamiento si ello resulta en una variación del número de recintos, del significado del identificador de recintos, o de la ubicación a tamaño de cada recinto dentro de la resolución del componente de losa. Los bins de datos del encabezamiento de losa de tren JPP y los bins de datos del encabezamiento principal se cargarán para las nuevas imágenes.

No se especifica cómo se forman los paquetes de recinto a partir de los bins de datos de recinto. El cliente no tiene por qué proporcionar todos los paquetes de un recinto, ni tampoco completar el último paquete proporcionado.

Al fusionar datos, el número de tamaño de recintos no debe variar y los datos que no se han reemplazado mediante el proceso de carga deberán tener el mismo significado que tenían antes de dicho proceso.

E.4.4 Bins de metadatos del tren JPP y del tren JPT

Pueden cargarse bins de metadatos que sustituyan el contenido de los bins de metadatos existentes. Dado que el servidor controla la división de la atribución de metadatos en los bins de datos, el cliente debe seguir la estructura de bins de metadatos del servidor. El cliente no cambiará los marcadores de posición en los bins de metadatos, salvo para suprimir completamente un marcador de posición. Al cargar un bin de metadatos entero, los clientes pueden añadir nuevos metadatos al final de los bins de metadatos, o insertar nuevos metadatos entre casillas en los bins de metadatos antiguos. El servidor gestionará los marcadores de posición y la estructura de bins de metadatos, lo que incluye la actualización de todos los marcadores de posición que apuntan a cualquier casilla de metadatos eliminada que haya sido modificada o afectada por el cambio. El servidor suprimirá todas las casillas de metadatos a las que apuntaba un marcador de posición suprimido por el cliente. El servidor puede reestructurar los metadatos después de aceptar la carga, pero antes de crear el nuevo recurso. Si después de la carga quedan secciones no usadas en el archivo, deberán utilizarse casillas de relleno para completar dichas secciones.

E.4.5 Carga de una imagen entera

En el caso de aceptar la carga de una imagen entera, el servidor debe descomprimir (si fuera necesario) la subimagen cargada, descomprimir alguna parte de la imagen total en el servidor, sustituir los píxeles en el dominio de espacio (no comprimido) y recomprimir todas las losas o recintos afectados por la operación de actualización.

NOTA – Si bien esta técnica exige una mayor capacidad de cálculo en el servidor, elimina la posibilidad de que el cliente utilice datos de imagen comprimida de manera incompatible (por ejemplo, un número de niveles incorrecto de la transformada en ondícula discreta).

Anexo F

Utilización de JPIP sobre HTTP

(Esta anexo es parte integrante de esta Recomendación | Normalización Internacional)

F.1 Introducción

El presente anexo define el método de utilizar JPIP junto con el HTTP para peticiones y respuestas. Los parámetros de petición JPIP del anexo C se encapsulan en estructuras de petición HTTP permitidas. Las respuestas del servidor (incluidos los códigos de estado, los encabezamientos, los mensajes y los códigos de respuestas) del anexo D se encapsulan en respuestas HTTP permitidas. Todas las peticiones y respuestas deben codificarse de conformidad con lo especificado en la norma HTTP.

Obsérvese que el texto y los ejemplos de este anexo describen la utilización de JPIP por HTTP. Se prevé que la misma vinculación pueda utilizarse para HTTP con seguridad.

F.2 Peticiones

F.2.1 Introducción a las peticiones

En el anexo C se definen los campos necesarios. Cuando se transporta a través del HTTP, la petición JPIP puede aparecer como una cadena de consulta de una petición "GET" HTTP o el cuerpo de una petición "POST" HTTP. Dado que algunos sistemas HTTP limitan la longitud de la cadena de consulta proporcionada en la petición "GET", para peticiones JPIP grandes se prefiere la petición "POST"

NOTA 1 – La petición Request HTTP se define en la sección 5 de RFC 2616 del modo siguiente:

```
Request = Request-Line           ; HTTP Section 5.1
        0*(( general-header      ; HTTP Section 4.5
           / request-header      ; HTTP Section 5.3
           / entity-header ) CRLF) ; HTTP Section 7.1
        CRLF
        [ message-body ]        ; HTTP Section 4.3
```

NOTA 2 – La Request-Line HTTP y Request-URI se definen así:

```
Request-Line = Method SP Request-URI SP HTTP-Version CRLF
Request-URI = "*" / absoluteURI / abs_path / authority
```

NOTA 3 – RFC 2396 define:

```
absoluteURI = scheme ":" ( hier_part / opaque_part )
hier_part   = ( net_path / abs_path ) [ "?" query ]
abs_path    = "/" path_segments
```

F.2.2 Petición GET

Una petición JPIP puede proporcionarse a un servidor en la forma de una petición HTTP. Para una petición "GET", la petición HTTP está restringida de la siguiente manera:

- El "Método" será "GET".
- La "consulta" será cero o varios campos `jpip-request-field` separados por "&".

Un ejemplo de petición JPIP encapsulado en una petición "GET" HTTP es:

```
GET /images/kids.jp2?rsiz=640,480&roff=320,240&fsiz=1280,1024 HTTP/1.1
Host: get.jpeg.org
CRLF
```

Un ejemplo equivalente que utiliza `absoluteURI` en lugar de `abs_path` es el siguiente:

```
GET http://get.jpeg.org/images/kids.jp2?rsiz=640,480&roff=320,240
&fsiz=1280,1024 HTTP/1.1
```

CRLF

NOTA – Esta Recomendación | Norma Internacional no impone restricciones sobre el componente del plan de `absoluteURI`.

F.2.3 Peticiones POST

Una petición JPIP puede proporcionarse a un servidor encapsulada en una petición "POST" HTTP. Para una petición "POST", la petición HTTP está restringida de la manera siguiente:

- El "Método" deberá ser "POST".
- El "cuerpo de la entidad" deberá ser cero o varios campos `jpip-request-field` separados por '&'.
- La línea de encabezamiento "Content-type:" debe incluirse como una "entity-header" y contener el valor "application/x-www-form-urlencoded".

Un ejemplo de petición JPIP encapsulada en una petición "POST" HTTP es el siguiente:

```
POST /cgi-bin/j2k_server.cgi HTTP/1.1
```

Host: post.jpeg.org

Content-type: application/x-www-form-urlencoded

Content-length: 62

CRLF

```
target=/images/kids.jp2&rsiz=640,480&roff=320,240&fsiz=1280,1024
```

F.2.4 Peticiones de carga

Una petición de carga es una petición HTTP permitida con las siguientes restricciones:

- El "Método" deberá ser "POST".
- El URL deberá contener el campo de consulta de carga.
- El Content-type deberá ser el tipo de imagen del cuerpo: `image/jpt-stream`, `image/jpp-stream`, o un tipo de medios de imagen completa.

Un ejemplo de petición de carga JPIP es:

```
POST /images/kids.jp2?rsiz=640,480&roff=320,240&fsiz=1280,1024 HTTP/1.1
```

Host: post.jpeg.org

Content-type: image/jpt-stream

CRLF

F.3 Establecimiento de una sesión

Una sesión HTTP basada en sesión se establece utilizando el campo petición de Nuevo Canal con el valor "http", es decir, "`cnew=http`", como parte de una petición. Esta petición se transmite normalmente por HTTP. La petición puede contener una petición de ventana seleccionada que se convierte en la primera petición en el nuevo canal. La respuesta a esta petición se devuelve por la misma conexión por la que se recibió la petición.

El cliente puede abrir una conexión HTTP y realizar una petición que incluya el encabezamiento HTTP "`Connection: keep-alive`". Esto es útil para sesiones eficientes, pero no es una condición necesaria ni suficiente para tener una sesión. Puede utilizarse una misma conexión HTTP para el tráfico de diferentes destinos, diferentes canales, o incluso tráfico no-JPIP, por ejemplo peticiones para ficheros HTML. Una petición de JPIP que forma parte de una sesión puede llegar por conexiones de HTTP distintas de la conexión HTTP utilizada para la petición y creación del nuevo canal, aunque esta opción no se recomienda.

F.4 Respuestas

F.4.1 Introducción

Cada componente de una respuesta del anexo D pueden encapsularse como una parte de una respuesta HTTP permitida.

NOTA – La Response HTTP se define en la sección 6 de RFC 2616, del modo siguiente :

```
Response = Status-Line           ; HTTP Section 6.1
          0* ( ( general-header   ; HTTP Section 4.5
              / response-header  ; HTTP Section 6.2
              / entity-header )  ; HTTP Section 7.1
            CRLF
          [ message-body ]       ; HTTP Section 7.2
```

Las respuestas JPIP transportados por HTTP serán respuestas HTTP conformes, con restricciones adicionales sobre algunas de las partes de la respuesta HTTP, como se describe en las siguientes subcláusulas.

F.4.2 Código de estado y descripción de la razón

Todos los códigos de estado enumerados en D.1.3 pueden utilizarse directamente como código de estado HTTP. Además, el servidor que ofrece JPIP a través de HTTP puede utilizar los códigos de estado HTTP que considere útiles, por ejemplo 402.

Todos los valores de la descripción de razón que figuran en D.1.3 pueden utilizarse directamente como descripción de razón HTTP. La descripción de razón debe ser adecuada al código de estado. El servidor que proporcione JPIP a través de HTTP puede utilizar cualquier descripción de razón HTTP que considere útil, por ejemplo, "se exige pago".

F.4.3 Información en el encabezamiento

F.4.3.1 Encabezamientos JPIP

Las líneas de encabezamiento de D.2 se incluirán en la "entity-header" de la respuesta HTTP sin modificación alguna.

F.4.3.2 Utilización del encabezamiento Accept HTTP

El servidor que proporciona JPIP a través de HTTP puede utilizar una línea de encabezamiento "Accept:" HTTP, que figura en una petición, para determinar el tipo de respuesta JPIP. Si la petición contiene el parámetro de consulta "type=", el tipo de retorno será uno de los tipos enumerados en el parámetro tipo. Si la petición contiene el parámetro de consulta "type=" y una línea de encabezamiento "Accept:", el servidor puede utilizar las prioridades especificadas en la línea "Accept:" para seleccionar uno de los tipos especificados en el parámetro de consulta "type=". Si la petición no contiene el parámetro de consulta "type=", el servidor puede seleccionar un tipo de retorno que soporte el servidor JPIP subyacente de la lista de tipos en la línea "Accept:".

F.4.3.3 Utilización del encabezamiento control de caché (Cache-Control)

Obsérvese que las cachés en los servidores intermediarios HTTP son diferentes de las cachés y de los modelos de caché en JPIP.

Toda petición JPIP con un campo petición de Nuevo Canal forma parte de la sesión y, en general, las respuestas no pueden almacenarse en caché mediante servidores intermediarios HTTP. Análogamente, toda respuesta que incluya un encabezamiento de respuesta Nuevo Canal también forma parte de la sesión. En ambos casos, la respuesta del servidor debe incluir la línea de encabezamiento "Cache-Control:" HTTP con el valor "no-cache".

F.4.3.4 Utilización del encabezamiento Content-type

El servidor que proporcione JPIP a través de HTTP debe incluir la línea de encabezamiento "Content-type:" que indica el tipo de datos en el cuerpo, lo cual es muy común si se trata de imagen/tren jpp o imagen/tren jpt.

F.4.3.5 Utilización del encabezamiento Redirect

El encabezamiento Redirect HTTP puede ser útil para informar al cliente de que los recursos se han desplazado o de que debe accederse al mismo a través de un computador diferente.

Obsérvese que la respuesta JPIP indica también cómo realizar el redireccionamiento. La respuesta JPIP debe tener preferencia dentro de una sesión.

F.4.4 Cuerpo

Los mensajes del anexo D se incluirán en el cuerpo de la respuesta HTTP. Obsérvese que la respuesta HTTP deberá tener un mecanismo para determinar la longitud de la misma. Si el servidor no tiene previsto interrumpir una respuesta,

puede proporcionar esta información en la línea de encabezamiento HTTP "Content-Length". El método recomendado para proporcionar la longitud es la utilización de la línea de encabezamiento HTTP "Transfer-Encoding: chunked" y posteriormente que el servidor proporcione el cuerpo en fragmentos de un determinado tamaño y especificados antes de cada fragmento. No se recomienda indicar el final de una respuesta mediante la terminación de las conexiones HTTP.

F.5 Características del HTTP adicionales

F.5.1 Utilización del método HEAD del HTTP

Los clientes y servidores JPIP no están obligados a utilizar o soportar el método "HEAD" del HTTP. El servidor que elija implementar dicho método, cumplirá lo especificado en la sección 9.4 de RFC 2616, a saber, "el método HEAD es idéntico a GET salvo que el servidor no devuelve un cuerpo de mensaje en la respuesta".

Los clientes pueden considerar útil emplear las peticiones "HEAD" del HTTP como mecanismo para determinar si el servidor modificará algunos de los parámetros de petición especificados en el anexo D. Los clientes no deben emplear una petición "HEAD" HTTP con los campos de consulta del modelo caché dado que esto puede causar que el servidor actualice su modelo de caché.

Obsérvese que si un cliente desea actualizar el modelo de caché del servidor sin recibir una respuesta puede emplear el campo petición de longitud máxima de la respuesta.

Los servidores pueden rechazar alguna o todas las peticiones "HEAD". A diferencia de las peticiones "HEAD" HTTP típicas que exigen relativamente poco esfuerzo al servidor para atenderlas, algunas implementaciones del servidor JPIP puede que tengan que obtener los datos de diferentes ubicaciones de un destino lógico, calcular la naturaleza de la respuesta y descargar el cuerpo de la respuesta a fin de responder a la petición "HEAD".

F.5.2 Utilización del método OPTIONS del HTTP

Los clientes y servidores JPIP no están obligados a utilizar o soportar el método "OPTIONS" del HTTP.

F.5.3 Utilización de Etag

Obsérvese que el HTTP define un mecanismo de etiquetado de identidades (ETag) que es similar al campo de petición y de destino JPIP en cuanto que se utiliza para indicar cambios en un recurso. Si un recurso tiene asignado una etiqueta de identidad y un ID de destino, se recomienda que se modifique el ETag definido por HTTP siempre que se modifique el `target-id`.

F.5.4 Utilización de codificación de transferencia en fragmentos

Dado que las respuestas que contienen datos comprimidos pueden ser muy grandes y por consiguiente tardar mucho tiempo en transmitirse, es importante poder detenerlas a mitad de la transmisión. A no ser que se especifique "Transfer-Encoding: chunked", las peticiones HTTP deben especificar la longitud total del cuerpo en el encabezamiento "Content-Length:" o indicar el final de los datos cerrando la conexión. Ninguna de estas dos opciones es conveniente en un protocolo interactivo, dado que podría ser necesario detener la respuesta en curso y enviar más datos en la misma conexión para una nueva respuesta.

NOTA 1 – En la sección 19.4.6 de RFC 2616 figura un algoritmo para suprimir la codificación de transferencia en fragmentos.

NOTA 2 – La codificación de transferencia en fragmentos puede ser útil en el JPIP cuando se transmite sobre protocolos distintos del HTTP.

F.6 HTTP y el campo de petición de longitud (informativo)

En un canal de retorno HTTP, el servidor no recibe información continua del cliente y fácilmente puede colocar un gran volumen de datos en la cola, que habrá de recibirse íntegramente antes de que se puedan procesar los datos correspondientes a una nueva ventana. Para mantener la capacidad de respuesta, los clientes deben utilizar el campo de petición de longitud máxima de respuesta para controlar el flujo de tráfico y por consiguiente mantener la capacidad de respuesta. Por lo general, los clientes necesitarán implementar sus propios algoritmos de control de flujo para ajustar la longitud de la petición a las condiciones cambiantes de la red.

Anexo G

Utilización de JPIP con peticiones HTTP y respuestas TCP (Este anexo es parte integrante de esta Recomendación | Norma Internacional)

G.1 Introducción

El protocolo JPIP es neutral en lo que respecta a los mecanismos de transporte subyacente para las peticiones de cliente y la respuesta del servidor, salvo en lo relativo a las peticiones de canal representadas mediante el campo petición de nuevo canal ("cnew") (véase C.3.3) y el encabezamiento de respuesta nuevo canal ("JPIP-cnew") (véase D.2.3), en las que habrá que comunicar los detalles específicos de transporte. En la presente Recomendación | Norma Internacional se definen dos tipos de transporte específicos, que se identifican mediante las cadenas "http" y "http-tcp" en la cadena de valor correspondiente a las peticiones de nuevo canal. En este anexo se describe el segundo transporte, que en este texto se denomina HTTP-TCP. El primer tipo de transporte se denomina HTTP y se describe en el anexo F.

El transporte HTTP-TCP utiliza exactamente los mismos mecanismos que el transporte HTTP para enviar peticiones del cliente al servidor y recibir del mismo los encabezamientos de respuesta y códigos de estado. Ahora bien, los datos de respuesta del servidor (no los encabezamientos de respuesta) se transmiten por una conexión TCP auxiliar. La información que transporta esta conexión TCP es idéntica a la que se transportaría en cuerpo de la entidad de una respuesta puramente HTTP, salvo que se divide en fragmentos, cada uno de los cuales tiene un número de secuencia de fragmento.

El cliente acusa recibo explícitamente de la llegada de cada fragmento enviando como respuesta al servidor el número de secuencia por el trayecto de retorno de la conexión TCP auxiliar. Una de las ventajas más importantes del transporte de HTTP-TCP es que el servidor recibe la notificación incremental de la llegada de sus fragmentos de datos de respuesta a través del mecanismo de acuse de recibo del cliente. Esto permite que el servidor gestione en flujo de datos de manera que puedan mantenerse la capacidad de respuesta y la eficiencia de la red.

Todas las peticiones que se envían a través del transporte HTTP deberán codificarse como se especifica en la norma HTTP.

G.2 Peticiones de cliente

Las peticiones se transmiten por el canal primario exactamente igual a las peticiones HTTP. Su forma es exactamente la misma a las peticiones que se transmiten por un canal que utiliza el transporte HTTP descrito en el anexo F. Concretamente, pueden utilizarse las peticiones "GET" y "POST" del HTTP.

G.3 Establecimiento de sesión

G.3.1 Establecimiento del canal

Es posible establecer un nuevo canal en un servidor JPIP mediante una petición que incluya el campo petición de nuevo canal (véase C.3.3). Por ejemplo, podrá realizarse una petición de este tipo utilizando HTTP, aunque también podría solicitarse a un servidor específico JPIP utilizando cualquiera de los mecanismos de transporte adecuados. Si la respuesta del servidor (a través del encabezamiento de respuesta de nuevo canal descrito D.2.3) indica que el nuevo canal se ha creado para funcionar con el transporte HTTP-TCP, el cliente establecerá la conexión TCP auxiliar utilizando el número de puerto auxiliar devuelto en el encabezamiento de respuesta nuevo canal. Además, la petición que incluye el campo petición de nuevo canal se gestiona como si hubiera sido transmitida por el nuevo canal de transporte HTTP-TCP recientemente creado, lo que significa que los datos de respuesta generados por la respuesta se devuelven a través de la conexión TCP auxiliar, tan pronto como se establezca la conexión.

Para establecer la conexión TCP auxiliar, el cliente transmite una petición de conexión TCP al ordenador servidor indicado en el encabezamiento de respuesta de nuevo canal, y por el puerto designado por el encabezamiento de respuesta de nuevo canal. Inmediatamente después, el cliente envía una sola línea de texto ASCII, que consiste en la cadena de identificador de nuevo canal, seguido de dos pares consecutivos CR-LF. Ésta es la única comunicación de texto que se transmite por la conexión TCP auxiliar.

A continuación, el cliente espera hasta recibir los datos de respuesta del servidor por la conexión TCP auxiliar. Estos datos de respuesta no pueden estar vacíos dado que cada petición realizada por el canal de transporte HTTP-TCP deberá tener un tren de datos de respuesta que conste de al menos el mensaje EOR (véase D.3). Para mayor información véase G.4.

G.3.2 División de los datos de respuesta por el servidor

Los datos de respuesta que envía el servidor a través de su conexión TCP auxiliar pueden dividirse en fragmentos. Cada fragmento consta de un encabezamiento del fragmento de 8 bytes, seguido del cuerpo del fragmento que contiene los datos de respuesta del servidor, como se muestra en la figura G.1. La primera palabra de 2 bytes del encabezamiento del fragmento contiene un entero sin signo big endian que representa la longitud total del fragmento, incluida la propia palabra de longitud. El contenido de 6 bytes restantes del encabezamiento del fragmento no se definen en esta Recomendación | Norma Internacional, y podría utilizarse para la señalización adicional específica del servidor. El cliente devolverá el encabezamiento del fragmento de 8 bytes entero en sus mensajes de acuse de recibo de fragmento.

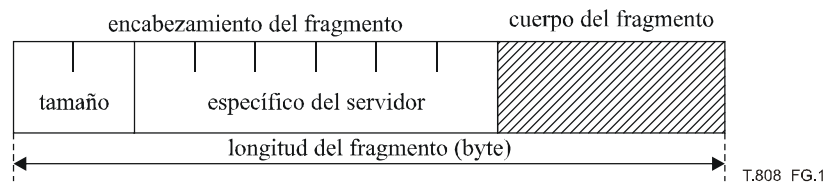


Figura G.1 – Estructura de datos de la respuesta por una conexión http-tcp

G.3.3 Acuse de recibo de los fragmentos de respuesta del servidor por el cliente

Tras recibir un fragmento de datos de respuesta del servidor a través de la conexión TCP auxiliar, el cliente responderá al servidor con un encabezamiento de fragmento de 8 bytes que será un tren de datos sin dividir, utilizando el trayecto retorno de la conexión TCP. Se acusa recibo secuencialmente de cada fragmento recibido.

G.4 Respuestas del servidor

En respuesta a cada petición del cliente, el servidor devuelve al mismo un párrafo de respuesta HTTP a través del canal primario. El párrafo de respuesta contiene el código de estado, la descripción de la razón y todos los encabezamientos de respuesta JPIP respectivos y los encabezamientos de respuesta HTTP adecuados. Ahora bien, no se devuelven datos de respuesta por el canal primario. Por esa razón, no habrá un cuerpo de entidad HTTP en la respuesta HTTP-TCP. Tampoco deberán utilizarse los encabezamientos de respuesta "Content-length:" o "Transfer-encoding:" HTTP.

Los datos de la respuesta se transmiten por el canal TCP auxiliar, dividido en fragmentos según lo descrito en G.3.2. Dado que el transporte HTTP-TCP puede utilizarse solamente en las sesiones y por consiguiente solamente con los tipos de retorno de imagen tren JPP y tren JPT, los datos de respuesta siempre constan de una secuencia de mensajes de tren JPP o tren JPT.

Los datos de la respuesta que resultan de cada petición consistirán en un número entero de fragmentos, lo que significa que ningún fragmento puede contener datos de respuesta generados en respuesta a dos peticiones diferentes.

La respuesta a todas y cada una de las peticiones terminarán con el mensaje EOR (véase D.3), aun cuando los datos de respuesta hubieran estado vacíos en caso contrario. El mensaje EOR se considera parte de los datos de respuesta y se divide en fragmentos junto con los mensajes de tren JPP y tren JPT.

Esto significa que cada petición que se transmite a través de un canal JPIP transportado por HTTP-TCP resulta en la generación de al menos un fragmento de respuesta no vacío del servidor, y que el último fragmento generado en respuesta a cada petición termina con el mensaje EOR.

Obsérvese que en realidad no existe el requisito de que los fragmentos de respuesta transportados por HTTP-TCP estén alineados en las fronteras del mensaje.

G.5 TCP y campo de petición de longitud (informativo)

Puede haber muy pocas razones o ninguna de utilizar el campo de petición de longitud máxima de la respuesta en un canal de retorno TCP, dado que el servidor es capaz de controlar meticulosamente el flujo de los datos de respuesta hacia el cliente y mantener así la capacidad de respuesta.

Anexo H

Utilización de JPIP con transportes diversos

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

H.1 Introducción

La presente Recomendación | Norma Internacional no define protocolos de transporte distintos del transporte "http" descrito en el anexo F y el transporte "http-tcp" descrito en el anexo G. El objetivo de este anexo es dar directrices sobre el despliegue de JPIP por transportes no fiables y proporcionar un método genérico que puede aplicarse a una gran variedad de transportes.

Al desarrollar el método general, resulta de ayuda dividir los aspectos de la comunicación en dos conexiones de transporte lógicas, denominadas "conexión de petición" y "conexión de datos". Se sobreentiende que cada conexión lógica proporciona un trayecto de comunicación directo y un trayecto de comunicación inverso. Las funciones de estos trayectos son las siguientes:

- el trayecto de conexión de la petición en sentido directo se utiliza para transmitir las peticiones JPIP desde el cliente al servidor;
- el trayecto de conexión de la petición inversa lo utiliza el servidor para acusar recibo de las peticiones recibidas y devolver los encabezamientos de respuesta al cliente;
- el trayecto de conexión de datos en sentido directo se utiliza para transmitir los mensajes del tren JPIP desde el servidor al cliente;
- el trayecto de conexión de datos en sentido inverso lo utiliza el cliente para acusar recibo de los mensajes del tren JPIP procedentes del servidor.

Cabe observar que estas funciones son compatibles con las de los trayectos de comunicación en sentido directo e inverso de los dos canales TCP que se utilizan en el transporte "http-tcp" descrito en el anexo G. Por otra parte, el material de este anexo puede interpretarse como una ampliación del transporte "http-tcp" para transporte no fiable. No obstante, obsérvese que aunque este anexo se describe para el caso de dos conexiones lógicas diferentes, no hay ninguna razón que impida que la comunicación se lleve a cabo a través de una única conexión de transporte.

Por último, se supone que cada conexión lógica proporciona uno de los dos siguientes tipos de servicio:

- a) un servicio orientado al tren fiable, por ejemplo el que ofrece el TCP;
- b) un servicio orientado a paquetes no fiables, por ejemplo el que ofrece UDP. En este caso, los paquetes pueden llegar en cualquier orden o no llegar, y el procedimiento de toma de contacto de los acuses de recibo se implementará explícitamente de modo que pueda determinarse si el paquete ha llegado correctamente.

En este anexo se tienen en cuenta dos casos. En el primer caso, el trayecto de conexión de petición ofrece un servicio orientado al tren fiable, pero el trayecto de conexión de datos no es fiable. En el segundo caso, los trayectos de petición de petición y datos no son fiables. Es conveniente estudiar estos dos casos en orden.

H.2 Peticiones fiables con datos no fiables

En esta subcláusula la conexión de petición es fiable, lo que significa que las peticiones llegan en orden y sin pérdidas al servidor, y las respuestas del servidor son recibidas por el cliente en orden y también sin pérdidas. En este caso, los campos de petición y los encabezamientos de respuesta pueden comunicarse exactamente como en el protocolo "http-tcp", y por esa razón se recomienda el HTTP para el transporte de peticiones y encabezamientos de respuesta. Un protocolo de transporte de este tipo podría, por ejemplo, denominarse "http-udp", aunque esos detalles quedan fuera del alcance de este anexo.

Los mensajes del tren JPIP, incluidos los mensajes EOR (véase D.3) se dividirán en paquetes y se entregarán por alguna conexión de datos no fiable (por ejemplo por UDP). El cliente acusará recibo de cada uno de esos paquetes enviando de vuelta al servidor el encabezamiento del paquete. De este modo, el servidor puede evaluar las condiciones de la red y determinar si conviene retransmitir el paquete. En caso de que varíe la ventana seleccionada del cliente, el servidor podría decidir no retransmitir un paquete del que no se haya acusado recibo.

Al crear protocolos de transporte de este tipo, conviene tener en cuenta las siguientes directrices generales:

- a) Cada petición debe incluir un campo de petición ID de petición (véase C.3.5).
- b) Para cada petición, habrá su correspondiente mensaje EOR, aun cuando no se envíen mensajes de tren JPIP en respuesta a la petición. Este requisito también se aplica en el caso del transporte "http-tcp".

- c) Cada paquete de conexión de datos construido por el servidor consistirá en un número entero de mensajes de tren JPIP y/o mensajes EOR. Además, el primer mensaje del tren JPIP en cada paquete deberá contener un encabezamiento completo, sin basarse en la repetición del identificador del tren de código o de los componentes de clase de código de un mensaje anterior.
- d) Todos los mensajes de tren JPIP (no necesariamente mensajes EOR) que figuran en el paquete de conexión de datos deberán pertenecer a la respuesta de una misma petición, y el correspondiente ID de petición deberá codificarse en el encabezamiento del paquete.
- e) Los mensajes EOR deben figurar al final de un paquete que transporta el mismo valor de ID de petición que la petición cuya respuesta está terminando, o en un bloque de uno o varios mensajes EOR consecutivos que figuran al principio del primer paquete siguiente al último paquete que transporta ese ID de petición. Esta política permite agrupar los mensajes EOR que corresponden a uno o varias respuestas vacías consecutivas (por ejemplo debido a peticiones reemplazadas) en el primer paquete de las respuestas no vacías subsiguiente.
- f) Además del valor ID de la petición, cada encabezamiento de paquete debe incluir un número de secuencia de paquete. El contador de secuencias de paquete se pone a cero para el primer paquete relacionado con un determinado valor ID de petición. Los siguientes paquetes con el mismo valor ID de petición tendrán números de secuencia consecutivos. Gracias a esta política, el cliente puede identificar mensajes EOR que pueden no haberse recibido debido a la pérdida de paquetes. Es importante que el cliente sea capaz de relacionar las peticiones con los datos de respuesta, a fin de sincronizar los efectos de los enunciados de manipulación del modelo de caché en el servidor con el estado de su propia caché.
- g) Los clientes deben acusar recibo de la recepción de cada paquete, para lo cual envían mensaje de acuse de recibo al servidor a través del trayecto de la conexión de datos de respuesta. Cada mensaje de acuse debe contener una duplicación del encabezamiento del paquete recibido, pero podría contener además información adicional. Queda a discreción del cliente agrupar mensajes de acuse de recibo en diversos paquetes cuando construya los paquetes de acuse. Ahora bien, una agrupación excesiva podría afectar a la fiabilidad en la que los servidores pueden calcular las estadísticas de la red.
- h) El servidor no está obligado a retransmitir los paquetes que no han recibido acuse de recibo y los clientes no deben esperar la transmisión de los paquetes que falten. Un servidor inteligente podría, por ejemplo, retransmitir los paquetes que no han recibido acuse de recibo en función de su importancia para la ventana seleccionada del caso.

H.3 Peticiones no fiables con datos no fiables

Esta subcláusula trata de los transportes para el caso en que las conexiones de petición y datos no son fiables. Las directrices para la conexión de datos son exactamente las mismas que las descritas en H.2 para el caso en que los datos se transmiten de manera no fiable. No obstante, con una conexión de petición no fiable es posible que una o varias peticiones se pierdan o lleguen desordenadas al servidor. El JPIP está bien dotado para gestionar esta situación, dado que los servidores tienen la libertad de reemplazar peticiones anteriores cuando llega una nueva petición.

Las directrices generales que figuran a continuación deben observarse cuando se gestionen peticiones no fiables, además de las enumeradas en H.2 para la desconexión de datos no fiables.

- a) Cada paquete de datos debe incluir un encabezamiento que identifique el valor del ID de petición.
- b) Cada paquete de petición debe incluir además un número de secuencia, que transporte la información suficiente para determinar si se han recibido todos los paquetes correspondientes a una determinada petición.
- c) En muchos casos, los servidores pueden hacer caso omiso de los paquetes de una petición que falten cuando se recibe una nueva petición. Para ello, el servidor sólo tiene que enviar mensajes EOR en la conexión de datos, lo que indica que la petición para la cual faltan datos fue reemplazada de inmediato. No hay necesidad alguna de enviar mensajes de acuse de recibo en respuesta a los paquetes de petición. Tampoco es necesario enviar encabezamientos de respuesta para responder a peticiones que se están reemplazando de inmediato dado que algunos o todos los paquetes de la petición se perdieron.
- d) Para cada petición que llegue íntegramente al servidor, éste debe enviar uno o más paquetes de respuesta que indiquen el ID de petición y que incluyan los encabezamientos de respuesta. Esto debe ser así aun cuando la petición llegue después de que se haya emitido la respuesta para cualquier petición subsiguiente (por ejemplo, porque algunos paquetes de la petición llegaron con excesivo retraso). De este modo el cliente dispone de un mecanismo para determinar si una petición importante fue recibida por el servidor.
- e) El servidor deberá procesar ciertos tipos de peticiones para evitar la pérdida de sincronización con el cliente. Las más importantes de éstas son las peticiones que incluyen los campos de manipulación del

modelo de caché restantes. Para que el servidor pueda detectar tales peticiones, sin tener que serializar completamente el tren de peticiones, los encabezamientos del paquete de petición deben incluir los dos campos siguientes:

- 1) Una bandera que indique si el paquete pertenece a una petición que debe procesarse antes de las subsiguientes peticiones.
- 2) El ID de petición correspondiente a la petición más reciente para la cual está puesta a 1 la bandera mencionada en e1.

Si el servidor no recibe uno o varios paquetes de una petición con la bandera e1 puesta a 1 (es decir, llegan peticiones que cumplen la condición e2 y falta la petición con el flag e1) deberá mantenerse en estado inactivo hasta que el cliente retransmita los paquetes.

H.4 Sintaxis de petición y respuesta

La sintaxis de petición y respuesta descrita en los anexos C y D debe respetarse cuando se diseñen nuevos transportes para el protocolo JPIP. Ahora bien, se permite desarrollar representaciones binarias equivalentes de diversos campos de petición y encabezamientos de respuesta.

H.5 Establecimiento de sesiones

El campo petición de nuevo canal (véase D.2.3) y el correspondiente encabezamiento de respuesta podría utilizarse para crear canales relacionados con los protocolos de transporte distintos del "http" y "http-tcp" que se describen normativamente en esta Recomendación | Norma Internacional. Para ello, los nuevos nombres del protocolo de transporte pueden registrarse en la junta de registros definida en el anexo J. El procedimiento para crear canales para nuevos transportes debe cumplir los mismos convenios generales que el "http-tcp". En particular, los encabezamientos de respuesta para la petición que crea el nuevo canal deben devolverse por el mismo transporte que fue utilizado para crear el canal, mientras que los datos de respuesta pueden entregarse utilizando el transporte del nuevo canal.

Anexo I

Indexación de ficheros JPEG 2000 para JPIP

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

I.1 Introducción (informativo)

En la Rec. UIT-T T.800 | ISO/CEI 15444-1:2004 y en otras normas se define una familia de formatos de fichero JPEG 2000. Esta familia utiliza una sintaxis común, cuyo elemento fundamental es el contenedor denominado casilla. En este anexo se definen nuevas casillas del formato de fichero que contienen información de indexación, cuya inclusión en los ficheros de la familia JPEG 2000 puede facilitar la utilización de dichos ficheros en sistemas JPIP, puesto que permite a los lectores de ficheros localizar dentro de los mismos los elementos que necesitan para reconstruir de manera incremental las imágenes.

En particular, estas casillas son útiles para:

- la implementación del protocolo JPIP en el servidor;
- el acceso por parte del cliente a una imagen distante utilizando un protocolo más simple que permite acceder a las gamas de byte especificadas en el fichero.

En el presente anexo se definen las casillas de índice correspondientes a la información del nivel de fichero y a la información de tren de código. Las casillas pueden clasificarse del modo siguiente:

- La supercasilla índice de tren de código (cidx) indexa la información del tren de código correspondiente a las clases principales de encabezamiento, encabezamiento de losa, bins datos de losa y recinto de los trenes JPP y JPT. Contiene una casilla buscador de tren de código (cptr) que apunta al tren de código indexado, una casilla declaración (Manifest) (manf) que resume el resto del contenido y casillas tabla de índices, que son la casilla tabla de índices de encabezamiento (mhix), la supercasilla tabla de índices de parte de losa (tpix), la supercasilla tabla de índices del encabezamiento de losa (thix), la supercasilla tabla de índices de paquete de recinto (ppix) y la supercasilla tabla de índices de encabezamiento de paquetes (phix). Las casillas tabla de índices corresponden a los diferentes tipos de datos de tren de código representados por las clases de bin de datos en el tren JPP y en el tren JPT definidos en el anexo A. Las casillas de tabla de índices que son supercasillas contienen casillas índice de matriz de fragmentos (faix) o tablas de índices de encabezamiento que enumeran los elementos del tren de código reales. Las supercasillas tabla de índices de encabezamiento, tabla de índices de paquete de recintos y tabla de índices de encabezamiento de paquete también contienen una casilla declaración.
- Las supercasillas índice de fichero (fidx) que indexan la información a nivel de fichero correspondiente a la clase bins de metadatos del tren JPP y del tren JPT. Salvo cuando indexan el nivel superior del fichero, en cuyo caso se denominan casilla índice de fichero raíz, contienen una casilla buscador de fichero (fptr) que apunta a la supercasilla indexada. También pueden contener casillas de intermediario (prxy) que representa el contenido del fichero o supercasilla indexado.
- La casilla buscador de índices (iptr) apunta al índice del fichero raíz, lo que permite detectar su ubicación.

En la figura I.1 se ilustra un ejemplo de fichero JPEG 2000 que contiene casillas índice JPIP:

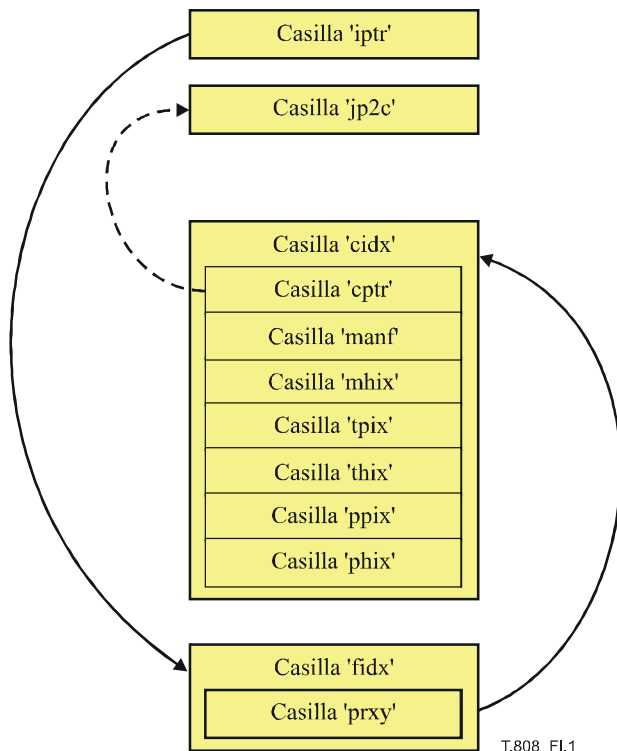


Figura I.1 – Parte de un fichero JPEG 2000 de ejemplo que contiene casillas índice JPIP

I.2 Identificación del uso de casillas índice JPIP en la lista de compatibilidad del formato de fichero JPEG 2000

Los ficheros que contienen una o varias casillas índice definidas en esta Recomendación | Norma Internacional pueden contener además un campo CL¹ en la casilla tipo del fichero (como se define en el anexo I a la Rec. UIT-T T.800 | ISO/CEI 15444-1) con el valor 'jpip' (0x6a70 6970).

I.3 Casillas definidas

I.3.1 Consideraciones generales

En el cuadro I.1 se enumeran todas las casillas definidas que forman parte de esta Recomendación | Norma Internacional. En cuanto a la colocación y las restricciones que se aplican a cada casilla, véase las correspondientes subcláusulas que la definen.

El cuadro I.1 es informativo. Las definiciones normativas de cada casilla figuran en cada una de las subcláusulas a las que se hace referencia en dicho cuadro.

Cuadro I.1 – Casillas definidas (informativo)

Nombre de la casilla	Tipo	Supercasilla	Comentarios
Casilla índice de tren codificado (I.3.2)	'cidx' (0x6369 6478)	Sí	Esta casilla contiene la información de índices relativa al tren de código JPEG 2000.
Casilla buscador de tren de códigos (I.3.2.2)	'cptr' (0x6370 7472)	No	Esta casilla apunta al tren de código JPEG 2000.
Casilla tabla de índices de encabezamiento (I.3.2.4.3)	'mhix' (0x6D68 6978)	No	Esta casilla especifica un índice de los segmentos marcadores en el encabezamiento principal de un tren de código o en los encabezamientos de parte de losa de una losa.
Casilla tabla de índices de parte de losa (I.3.2.4.4)	'tpix' (0x7470 6978)	Sí	Esta casilla especifica la ubicación y longitud de cada parte de losa en el tren de código.
Casilla tabla de índices de encabezamiento de losa (I.3.2.4.5)	'thix' (0x7468 6978)	Sí	Esta casilla especifica la ubicación y longitud de cada parte del tren de códigos necesario para construir los encabezamientos de losa de cada losa a fin de decodificar correctamente los datos de paquetes de recinto.
Casilla tabla de índices de paquete de recinto (I.3.2.4.6)	'ppix' (0x7070 6978)	Sí	Esta casilla especifica la ubicación y longitud de los paquetes dentro del tren de código.
Casilla tabla de índices de encabezamiento de paquete (I.3.2.4.7)	'phix' (0x7068 6978)	Sí	Esta casilla especifica la ubicación y longitud de los encabezamientos de paquete dentro del tren de código.
Casilla declaración (I.3.2.3)	'manf' (0x6D61 6E66)	No	Esta casilla resume las casillas que figuran inmediatamente después y contiguas a la misma, dentro de la casilla o fichero que la contiene en el mismo nivel que la casilla declaración.
Casilla índice de matriz de fragmento (I.3.2.4.2)	'faix' (0x6661 6978)	No	Esta casilla especifica la ubicación y longitud de los elementos del tren de código.
Casilla índice de fichero (I.3.3)	'fidx' (0x6669 6478)	Sí	Esta casilla puede utilizarse para buscar otros índices y datos cualesquiera dentro del fichero.
Casilla buscador de fichero (I.3.3.2)	'fptr' (0x6670 7472)	No	Esta casilla apunta a la casilla indexada.
Casilla intermediaria (I.3.3.3)	'prxy' (0x7072 7879)	No	Esta casilla representa en una casilla de índice de fichero a una casilla ubicada en otra parte del fichero.
Casilla buscador de índices (I.3.4)	'iptr' (0x6970 7472)	No	Esta casilla apunta a la casilla índice del fichero raíz de un fichero.

I.3.2 Casilla de índice del tren de código (supercasilla)

I.3.2.1 Descripción general

La casilla índice de tren de código contiene la información de índice relativa al tren de código JPEG 2000. El tipo de casilla de índice de tren codificado es 'cidx' (0x6369 6478). El contenido de la casilla de índice de tren codificado será el siguiente (figura I.2):

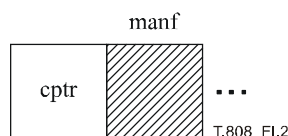


Figura I.2 – Organización del contenido de una casilla de índice de tren de código

- cptr:** Casilla buscador de tren de código. Esta casilla apunta al tren de código indexado por la casilla índice de tren de código. Su estructura se especifica en I.3.2.2.
- manf:** Casilla declaración. Esta casilla resume las tablas de índices que figuran después de la misma dentro de la casilla índice de tren de código. Su estructura se especifica en I.3.2.3.

I.3.2.2 Casilla buscador de tren de código

La casilla buscador de tren de código apunta al tren de código JPEG 2000. El tipo de casilla buscador de tren de códigos es 'cptr' (0x6370 7472). El contenido de la casilla buscador de tren de código será el siguiente (figura I.3):

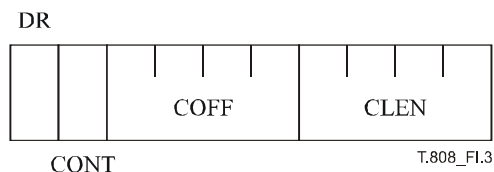


Figura I.3 – Organización del contenido de una casilla buscador de tren de código

- DR:** Referencia de datos. Este campo especifica la ubicación del tren de código, o de la casilla de tabla de fragmentos que la representa. Si su valor es 0, el tren de código o su casilla de tabla de fragmentos existe en el fichero del caso. De lo contrario, su valor indica un elemento de la casilla de referencia de datos en el fichero en cuestión. En este caso, el elemento de la referencia de datos identificado por la DR indica el recurso que contiene el tren de código o la casilla de tablas de fragmentos. Este campo se almacena en un entero sin signo big endian de 2 bytes.
- CONT:** Tipo de contenedor. Este campo se almacena en un entero sin signo big endian de 2 bytes. Los valores definidos en esta Recomendación | Norma Internacional se describen en el cuadro I.2.
- COFF:** Traslación del tren de código. Este campo especifica la ubicación del tren de códigos o de la casilla de lista de fragmentos, según corresponda, con respecto al principio del fichero o recurso identificado por la DR. Este campo se almacena en un entero sin signo big endian de 8 bytes.
- CLEN:** Longitud del tren de código. Este campo especifica la longitud del tren de código o de la casilla lista de fragmentos, según corresponda. El campo se almacena en un entero sin signo big endian de 8 bytes.

Cuadro I.2 – Valores del tipo de contenedor

CONT	Significado
0	El tren de código en su totalidad aparece como una serie continua de bytes dentro del fichero o recurso. En este caso, los valores de la traslación y longitud que se indican se refieren al propio tren de código. Obsérvese que el tren de código puede estar dentro de una casilla tren de código contiguo, pero los valores de traslación y longitud se refieren al propio tren de código, que comienza en el marcador SOC y termina inmediatamente después del marcador EOC.
1	El tren de código está fragmentado y los valores de ubicación y longitud se refieren a la casilla lista de fragmentos (incluido su encabezamiento de casilla) que describen la ubicación y longitud de cada uno de los fragmentos que representa el tren de código. Obsérvese que todas las ubicaciones y longitudes subsiguientes se expresan con respecto al principio del tren de código, tal como aparecería después de reconstituir todos los fragmentos identificados en la casilla lista de fragmentos.
Otros valores	Reservado para uso ISO.

I.3.2.3 Casilla declaración

La casilla declaración resume las casillas que figuran inmediatamente después y contiguas a la misma, dentro de su casilla o fichero que la contiene en el mismo nivel que la casilla declaración.

NOTA – La casilla declaración puede emplearse para facilitar el acceso aleatorio a las siguientes casillas, tales como las casillas índice siguientes a la misma dentro de la casilla índice de tren de código.

El tipo de casilla declaración es 'manf' (0x6D61 6E66). El contenido de la casilla declaración será el siguiente (figura I.4):

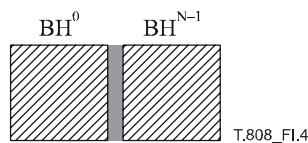


Figura I.4 – Organización del contenido de una casilla declaración

- BHⁱ:** Encabezamiento de casilla. Este campo contiene el encabezamiento de casilla íntegro de la *i*-ésima casilla que figura inmediatamente después de la casilla declaración. La longitud de este campo es de

16 bytes si el valor del campo LBox que figura dentro de ese encabezamiento de casilla es 1, u 8 bytes en caso contrario.

El número de casillas, (N) cuyos encabezamientos figuran dentro de la casilla declaración, viene dado por la longitud de la casilla declaración. Cuando se utiliza dentro de una casilla tabla de índices de paquete de recintos o de una casilla tabla de índices de encabezamiento de paquete, N es el número de componentes del tren de código.

Dentro de una casilla índice de tren de código, una casilla tabla de índices de encabezamiento de losa, una casilla tabla de índices de paquetes de recinto o una casilla tabla de índices de encabezamiento de paquete, la casilla declaración incluirá todas las casillas que siguen, hasta el final de la casilla que la contiene.

I.3.2.4 Tablas de índices

I.3.2.4.1 Descripción general

La casilla índice de tren de código puede contener una tabla de índices para cada uno de los siguientes tipos de datos de tren de código: encabezamiento principal, partes de losa, encabezamientos de losa, paquetes (recintos) y encabezamientos de paquetes. Cada tabla de índices es de un tipo de casilla diferente. No podrá haber más de una de cada tipo de tabla en la casilla de índice de tren de código.

Las casillas tabla de índice de parte de losa, tabla de índices de paquete de recinto y tabla de índices de encabezamiento de paquete son supercasillas que contienen casillas índice de matriz de fragmentos. La casilla tabla de índices de encabezamiento de losa es una supercasilla que contiene casillas tabla de índices de encabezamiento. A continuación, se define en primer lugar la casilla índice de matriz de fragmentos y después las casillas tabla de índices.

I.3.2.4.2 Casilla índice de matriz de fragmentos

La casilla índice de matriz de fragmentos contiene la ubicación y longitud de los elementos de un tren de código. Esta casilla se utiliza dentro de las supercasillas tabla de índices de parte losa, tabla de índices de paquetes de recinto y tabla de índices de encabezamiento de paquete.

El tipo de la casilla índice de matriz de fragmentos es 'faix' (0x6661 6978). El contenido de la casilla índice matriz de fragmentos será el siguiente (figura I.5):

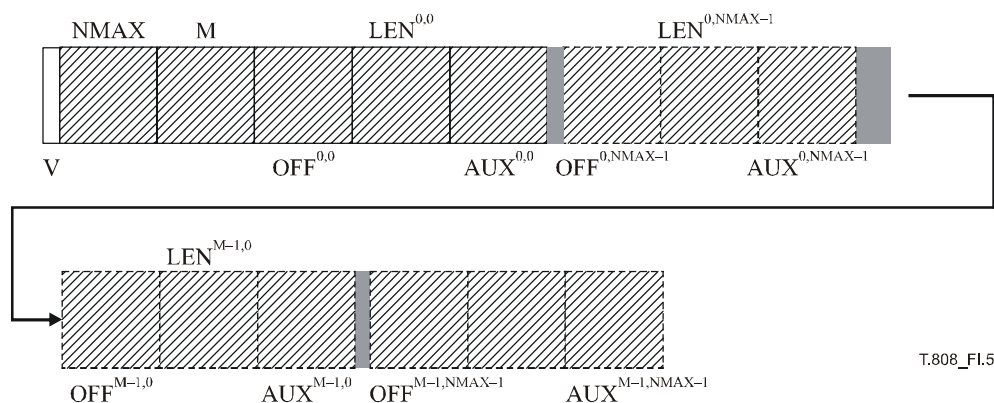


Figura I.5 – Organización del contenido de una casilla índice matriz de fragmentos

- V:** Versión. Este campo se codifica mediante un entero sin signo de 1 byte. Los valores definidos en esta Recomendación | Norma Internacional se describen en el cuadro I.3.
- NMAX:** Número máximo de elementos válidos en cada fila de la matriz. Cuando se utiliza dentro de una tabla de índices de tren codificado, NMAX es el número máximo de elementos que se especificarán para cualquier losa.
- M:** Número de filas de la matriz. Cuando se utiliza dentro de la tabla de índices del tren de código, M es el número de losas.
- OFF^{ij}:** Traslación. Este campo especifica la traslación en bytes (con respecto al comienzo del tren de código) del j -ésimo elemento en la fila i de la matriz.
- LEN^{ij}:** Longitud. Este campo especifica la longitud en bytes del j -ésimo elemento en la fila i de la matriz.
- AUX^{ij}:** Auxiliar. Este campo especifica la información auxiliar relativa al j -ésimo elemento de la fila i de la matriz. El valor de este campo será cero a no ser que la supercasilla que contiene esta casilla permita lo contrario. Todos los valores distintos de cero de este campo están reservados.

Si bien todas las casillas de la matriz especificada en la casilla de índice de matriz de fragmentos se almacenarán con NMAX número de elementos, el objeto que describe dicha fila puede especificar un número de elementos menor. En ese caso, para toda fila i que contenga J elementos válidos, siendo J menor que NMAX, los valores de $OFF^{i,j}$ a $OFF^{i,NMAX-1}$ y $LEN^{i,j}$ a $LEN^{i,NMAX-1}$ se pondrán a cero.

Cuadro I.3 – Valores de la versión

CONT	Significado
0	Los campos NMAX, M y todos los $OFF^{i,j}$ y $LEN^{i,j}$ se codifican como enteros sin signo big endian de 4 bytes, y no figuran campos $AUX^{i,j}$.
1	Los campos NMAX, M y todos los $OFF^{i,j}$ y $LEN^{i,j}$ se codifican como enteros sin signo big endian de 8 bytes y no hay campos $AUX^{i,j}$ presentes.
2	Todos los campos distintos de V se codifican como enteros sin signo big endian de 4 bytes.
3	Los campos NMAX, M y todos los $OFF^{i,j}$ y $LEN^{i,j}$ se codifican como enteros sin signo big endian de 8 bytes y todos los campos $AUX^{i,j}$ se codifican como enteros sin signo big endian de 4 bytes.
Otros valores	Reservado para uso ISO.

I.3.2.4.3 Casilla tabla de índice de encabezamiento

La casilla tabla de índices de encabezamiento indexa el encabezamiento principal de un tren de código o los encabezamientos de parte de losa de la losa, e indica la longitud total del encabezamiento principal o la longitud de la primera parte de losa y la ubicación y longitud de los segmentos marcadores en el encabezamiento. Todos los segmentos marcadores deberán incluirse, salvo el segmento marcador SOT que puede omitirse para los encabezamientos de parte losa que consistan únicamente de SOT y SOD. No es necesario enumerar los segmentos marcadores en el orden en que aparece en el tren de código. La casilla tabla de índices de encabezamiento sólo puede aparecer dentro de la casilla índice de tren de código. En el nivel superior, dicha casilla indexa un tren de código y no debe aparecer más de una vez. Dentro de la casilla tabla de índices del encabezamiento de losa, dicha casilla indexa los encabezamientos de parte de losa.

NOTA – Lo que se pretende es proporcionar un mecanismo eficaz para evitar la información de puntero en el encabezamiento, que no es imprescindible para navegar eficazmente por el fichero y que podría sobrecargar innecesariamente el encabezamiento. La lista de múltiples segmentos marcadores con el mismo código de marcador contiguo en la casilla de tabla de índices de encabezamiento permitirá a los lectores saltarse grupos de segmentos marcadores con los que no están interesados.

El tipo de casilla tabla de índices de encabezamiento es 'mhix' (0x6D68 6978). El contenido de la casilla tabla de índices de encabezamiento será el siguiente (figura I.6):

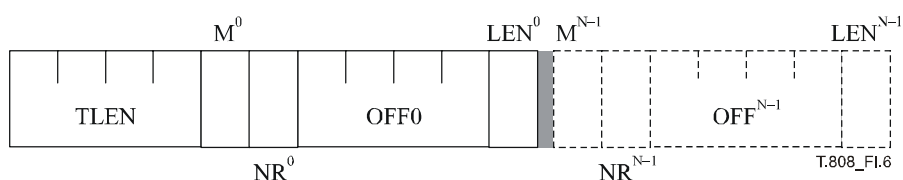


Figura I.6 – Organización del contenido de una casilla de tabla de índice de encabezamiento

- TLEN:** Longitud. Cuando la casilla tabla de índice de encabezamiento indexa un encabezamiento principal, este campo especifica la longitud total del encabezamiento principal. Cuando la casilla de tabla de índices de encabezamiento indexa un encabezamiento de parte de losa, este campo especifica la longitud total del primer encabezamiento de parte de losa. El valor de este campo se codifica como un entero sin signo big endian de 8 bytes.
- Mⁱ:** Código marcador. Este campo especifica el código marcador que comienza con el i -ésimo segmento marcador que figura en esta casilla. El valor de este código se codifica como un entero sin signo big endian de 2 bytes.
- NRⁱ:** Número restante. Este campo indica que (como mínimo) figuran, inmediatamente después y de manera contigua, NRⁱ segmentos marcadores con el mismo código marcador Mⁱ, a continuación del i -ésimo segmento marcador de esta lista. El valor de este campo se codifica con un entero sin signo big endian de 2 bytes.
- OFFⁱ:** Traslación. Este campo especifica la translación en bytes, con respecto al principio del tren de código, de los parámetros de segmento marcador (incluido el parámetro longitud pero sin contar el propio marcador) para el i -ésimo segmento marcador de la lista. El valor de este campo se codifica con un entero sin signo big endian de 8 bytes.

LENⁱ: Longitud. Este campo especifica la longitud en bytes de los parámetros del segmento marcador (incluido los dos bytes del parámetro longitud pero sin contar los bytes del propio marcador) para el *i-ésimo* segmento marcador de la lista. El valor de este campo se codifica con un entero sin signo big endian de 2 bytes, y su valor es el mismo que el del parámetro longitud del propio segmento marcador.

El número de segmentos marcadores (N), que figuran en la casilla de tabla de índices de encabezamiento viene dado por la longitud de la casilla tabla índices de encabezamiento.

I.3.2.4.4 Casilla tabla de índice de parte de losa (supercasilla)

La casilla tabla de índice de parte de losa indexa la ubicación y longitud de cada parte de losa en el tren de código, de modo que cada parte de losa comienza con su segmento marcador y termina con el último paquete de la parte de losa.

El tipo de casilla tabla de índice de parte losa es 'tpix' (0x7470 6978). El contenido de la casilla tabla índice de parte losa será el siguiente (figura I.7):

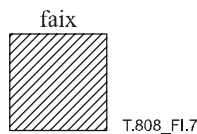


Figura I.7 – Organización del contenido de la casilla tabla de índice de parte losa

faix: Casilla índice de matriz de fragmentos. Esta casilla contiene la ubicación y longitud de todas las partes de losa en el tren de código. Su estructura se especifica en I.3.2.4.2. La *m-ésima* fila de esta tabla corresponde con la *m-ésima* losa en el tren de código. Los elementos de esta fila contienen la ubicación y longitud de todas las partes losa de la correspondiente losa, en el orden en que aparecen en el tren de código. Si la versión de la casilla de índice de matriz de fragmentos es igual a 2 ó 3, los campos auxiliares especifican para cada parte losa el n más pequeño, para todos los componentes para los que la diferencia ($N_L - n$) no es negativa, para el que se cumple que el nivel de resolución ($N_L - n$) y todos los niveles de resolución inferiores se han completado cuando esta parte losa se combina con todas las partes losa precedentes de la misma losa, siendo N_L el número de niveles de descomposición, que puede variar según el componente. Si ningún nivel de resolución de cualquier componente se ha completado, el valor del campo auxiliar es uno más el valor máximo de N_L a lo largo de todos los componentes. El valor cero corresponde al caso en que todas las resoluciones en todos los componentes se han completado. Dado que las resoluciones no tienen por qué aparecer en orden en una losa, pueden haberse completado algunos niveles de resolución anteriores al señalado por el campo auxiliar.

I.3.2.4.5 Casilla tabla de índice de encabezamiento de losa (supercasilla)

La casilla tabla de índice de encabezamiento de losa indexa los encabezamientos de losa de cada losa, para la correcta decodificación de los datos de paquete de recinto.

El tipo de casilla tabla de índice de encabezamiento de losa es 'thix' (0x7468 6978). El contenido de la casilla índice de encabezamiento de losa será el siguiente (figura I.8):

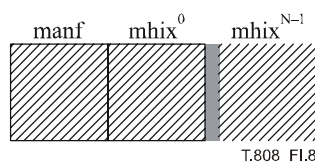


Figura I.8 – Organización del contenido de una casilla tabla de índice de encabezamiento de losa

El número de casillas tabla de índices de encabezamiento (N) es el número de losas.

manf: Casilla declaración. Esta casilla resume las casillas especificadas por $mhix^i$ dentro de su casilla de tabla de índices de encabezamiento de losa. Su estructura se especifica en I.3.2.3.

mhixⁱ: Casilla tabla de índice de encabezamiento. Esta casilla indexa los encabezamientos de parte de losa para la *i-ésima* losa. Su estructura se especifica en I.3.2.4.3.

I.3.2.4.6 Casilla tabla de índice de paquetes de recinto (supercasilla)

La casilla tabla de índice de paquetes de recinto indexa los paquetes dentro del tren codificado. El tipo de casilla tabla de índices de paquete de recinto será 'ppix' (0x7070 6978). El contenido de la casilla tabla de índice de paquetes de recinto será el siguiente (figura I.9):

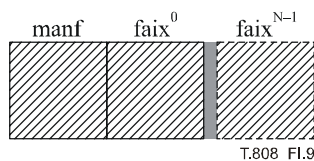


Figura I.9 – Organización del contenido de una casilla tabla de índice de paquetes recinto

El número de casillas índice de matriz de fragmentos, no será mayor que el número de componentes del tren de código.

- manf:** Casilla declaración. Esta casilla resume las casillas especificadas por $faix^i$ dentro de la casilla tabla de índice de paquetes de recinto. Su estructura se especifica en I.3.2.3.
- faixⁱ:** La *i*-ésima casilla índice de matriz de fragmentos corresponde a la *i*-ésima componente de la imagen en el tren de código. La *m*-ésima fila en esta tabla corresponde a la *m*-ésima losa en el tren de código. Los elementos de esta fila contienen la ubicación y longitud de todos los paquetes en la correspondiente componente losa. Los paquetes aparecen de manera contigua, en orden ascendente de capa, con sus respectivos recintos, y los recintos aparecen en el orden relativo al número de secuencia *s*, definido en A.3.2.1. No obstante, el orden fijo de los paquetes no tiene por qué ser el mismo que el especificado en cualquier segmento marcador COD/POC dentro del tren de código. La estructura de la casilla índice de matriz de fragmentos se especifica en I.3.2.4.2.

Si los encabezamientos de paquete se empaquetan dentro de segmentos marcadores PPM o PPT, los correspondientes elementos en la matriz de fragmentos se refieren a la ubicación y longitud del cuerpo del paquete únicamente, tal y como figuran dentro del cuerpo de la parte losa. Los elementos que hacen referencia a paquetes no existentes (porque la correspondiente componente losa contiene menos paquetes que otra componente losa en la misma matriz o porque el tren de código se ha truncado antes del punto en el que debería estar ese paquete) el correspondiente campo de ubicación debe ponerse a cero. Los elementos que hacen referencia a paquetes cuyo cuerpo está vacío y cuyo encabezamiento consta de exactamente un byte (0x80) pueden identificarse mediante una longitud de valor cero. Este tipo de paquetes aparecen frecuentemente en los trenes de código JPEG 2000; las aplicaciones deben evitar la sobrecarga que supone la búsqueda explícita del valor de dichos paquetes cuyo contenido es predecible. Si el correspondiente segmento marcador COD especifica que los marcadores EPH van a aparecer después de cada encabezamiento de paquete en alguna losa, una longitud de valor cero se interpretará para esa losa como que el paquete está formado por el byte 0x80 seguido del marcado EPH.

I.3.2.4.7 Casilla tabla de índice de encabezamiento de paquete (supercasilla)

La casilla tabla de índice de encabezamiento de paquetes indexa los encabezamientos de paquete dentro del tren codificado. El tipo de la casilla tabla de índice de encabezamiento de paquetes es 'phix' (0x7068 6978). El contenido de la casilla tabla de índices de encabezamiento de paquete será el siguiente (figura I.10):

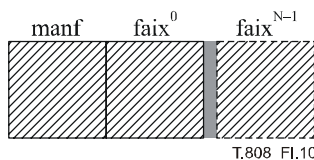


Figura I.10 – Organización del contenido de la casilla tabla de índice de encabezamiento de paquete

El número de casillas índice de matriz de fragmentos (*N*) no será mayor que el número de componentes del tren de código.

- manf:** Casilla declaración. Esta casilla resume las casillas especificadas mediante $faix^i$ dentro de la casilla tabla de índice de encabezamiento de paquete. Su estructura se especifica en I.3.2.3.
- faixⁱ:** La *i*-ésima casilla de índices de matriz de fragmento corresponde a la *i*-ésima componente de imagen en el tren de código. La *m*-ésima fila en esta tabla corresponde a la *m*-ésima losa en el tren

de código. Los elementos en esta fila contienen la ubicación y longitud de todos los encabezamientos de paquete en la correspondiente componente de losa. Los encabezamientos de paquete figuran continuamente, en orden ascendente de capa, dentro de sus respectivos recintos, y los recintos aparecen en el orden relativo al número de secuencia, definido en A.3.2.1. Ahora bien, el orden fijo de los encabezamientos de paquete no tiene por qué ser el mismo que el especificado por los segmentos marcadores COD/POC dentro del tren de código. La estructura de la casilla índice de matriz de fragmentos se especifica en I.3.2.4.2.

Los elementos que hacen referencia a encabezamientos de paquete no existentes (porque la correspondiente componente losa contiene menos paquetes que otra componente losa en la misma matriz, o porque el tren de código se ha truncado antes del punto en el que debía estar el encabezamiento del paquete) el correspondiente campo de aplicación debe ponerse a cero. Los elementos que hacen referencia a paquetes cuyo cuerpo está vacío y cuyo encabezamiento consta de exactamente 1 byte (0x80) pueden identificarse utilizando una longitud de valor cero. Este tipo de paquete aparece con frecuencia en los trenes de código JPEG 2000; las aplicaciones pueden evitar la sobrecarga que supone la búsqueda explícita de dichos paquetes cuyo contenido es predecible. Si el correspondiente segmento marcador COD especifica que los marcadores EPH van a aparecer después de cada encabezamiento de paquete en alguna losa, una longitud de valor cero se interpretará para esa losa como que el paquete está formado por el byte 0x80 seguido del marcador EPH.

I.3.3 Casilla índice de fichero (supercasilla)

I.3.3.1 Descripción general

La casilla índice de fichero puede utilizarse para buscar otros índices (en particular el índice del tren de código correspondiente al tren de código) y datos cualesquiera dentro del fichero.

La casilla índice de fichero raíz indexa el nivel superior del fichero. Cualquier otra casilla de índice de fichero indexa una supercasilla dentro del fichero. Como mucho habrá una sola casilla índice de fichero con un determinado alcance (nivel superior o una determinada supercasilla) dentro de un fichero dado.

El tipo de casilla de índice de fichero será 'fidx' (0x6669 6478). El contenido de la casilla índice de fichero será el siguiente (figura I.11):

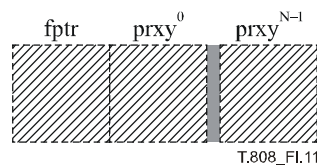


Figura I.11 – Organización del contenido de una casilla índice de fichero

- fptr:** Casilla buscador de fichero. La casilla índice de fichero raíz no incluirá esta casilla. Las demás casillas índice de fichero incluirán esta casilla, que apuntará a la supercasilla indexada por la casilla índice de fichero. La estructura de la casilla buscador de fichero se define en I.3.3.2.
- prxyⁱ:** Casilla intermediario. Esta casilla representa una casilla en la parte del fichero indexada por la casilla índice de fichero. La casilla índice de fichero raíz incluye intermediarios únicamente para las casillas en el nivel superior del fichero. Las demás casillas índice de fichero deberán incluir intermediarios sólo para las casillas en el nivel superior de la supercasilla indexada por la casilla índice de fichero. Los intermediarios aparecerán en el mismo orden que las casillas, aunque no todas las casillas necesitan un intermediario. La estructura de la casilla intermediaria se define en I.3.3.3.

NOTA – Dado que en algunos casos la presencia, ausencia u orden de las casillas en el fichero es significativo, para algunas aplicaciones sería útil que antes de las casillas con intermediarios, no se omitan casillas en el índice, dentro del alcance del mismo.

I.3.3.2 Casilla buscador de fichero

La casilla buscador de fichero apunta a una casilla. El tipo de casilla buscador de fichero es 'fptr' (0x6670 7472). El contenido de la casilla de buscador de fichero será el siguiente (figura I.12):

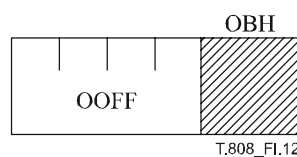


Figura I.12 – Organización del contenido de una casilla buscador de fichero

- OOFF:** Traslación original. Este campo especifica la traslación en bytes (relativa al comienzo del fichero) de la casilla a la que apunta esta casilla buscador de fichero. El valor de este campo se codifica con un entero sin signo big endian de 8 bytes.
- OBH:** Encabezamiento de la casilla original. Este campo contiene el encabezamiento de casilla íntegro de la casilla a la que apunta esta casilla de buscador de fichero. La longitud de este campo es de 16 bytes si el valor del campo LBox que contiene ese encabezamiento de casilla es 1, o de 8 bytes en caso contrario.

I.3.3 Casilla intermediario

En una casilla índice de fichero, la casilla intermediario representa una casilla ubicada en cualquier lugar del fichero, e indica su ubicación y longitud, la ubicación y longitud de todo índice de la casilla y un prefijo del contenido de la casilla.

El tipo de casilla intermediario será 'prxy' (0x7072 7879). El contenido de la casilla intermediario será el siguiente (figura I.13):

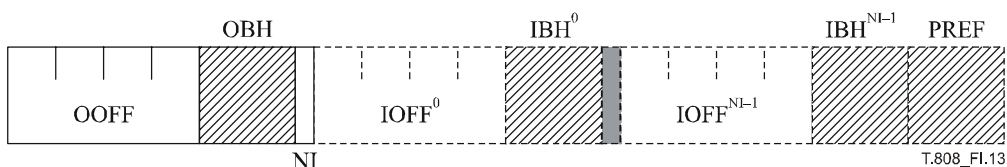


Figura I.13 – Organización del contenido de una casilla intermediario

- OOFF:** Traslación original. Este campo especifica la traslación en bytes (con respecto al principio del fichero) de la casilla representada por esta casilla intermediario. El valor de este campo se codifica con un entero sin signo big endian de 8 bytes.
- OBH:** Encabezamiento de casilla original. Este campo contiene el encabezamiento de casilla íntegro de la casilla representada por esta casilla intermediario. La longitud de este campo es de 16 bytes si el valor del campo LBox que contiene ese encabezamiento de casilla es 1, u 8 bytes en caso contrario.
- NI:** Número de índices. Este campo indica el número de punteros índice que incluye la casilla intermediario. Cada conjunto de campos IOFFⁱ, e IBHⁱ subsiguientes apunta a una casilla índice de fichero o índice de tren de código que indexa la casilla representada por esta casilla intermediario. Todos los demás valores están reservados. Los valores de este campo se codifican con un entero sin signo de 1 byte.
- IOFFⁱ:** Traslación de índice. Este campo contiene la traslación en bytes (con respecto al principio del fichero) de la *i-ésima* casilla índice. El valor de este campo se codifica con un entero sin signo big endian de 8 bytes.
- IBHⁱ:** Encabezamiento de casilla de índice. Este campo contiene el encabezamiento de casilla completo de la *i-ésima* casilla índice. La longitud de este campo es de 16 bytes si el valor del campo LBox que contiene ese encabezamiento de casilla es 1, u 8 bytes en caso contrario.
- PREF:** Prefijo. Este campo contiene un prefijo arbitrario de los datos en la casilla representados por esta casilla intermediario. El valor de su longitud puede estar entre cero y la longitud del contenido de la casilla original.

I.3.4 Casilla buscador de índice

La casilla buscador de índice apunta a la casilla índice de fichero raíz de un fichero. Sólo aparecerá si el fichero contiene una casilla índice fichero raíz. El tipo de casilla buscador fichero es 'iptr' (0x6970 7472). El contenido de una casilla buscador fichero será el siguiente (figura I.14):

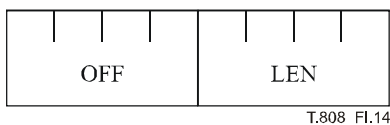


Figura I.14 – Organización del contenido de la casilla buscador de fichero

- OFF:** Traslación. Este campo especifica la ubicación de la casilla índice de fichero raíz con respecto al principio del fichero. Este campo se almacena en un entero sin signo big endian de 8 bytes.

LEN: Longitud. Este campo especifica el tamaño de la casilla índice de fichero raíz. Este fichero se almacena en un entero sin signo big endian de 8 bytes.

I.4 Relación de los índices de trenes de código con los trenes de código

En un fichero JP2, JPX o JPM, la casilla índice de tren de código aparecerá en el nivel superior del fichero, y la *i-ésima* casilla índice de tren de código corresponderá al *i-ésimo* tren de código, también en el nivel superior del fichero. La casilla buscador de tren de código dentro de la casilla índice de tren de código también indicará el tren de código que está indexado por la casilla índice de tren de código.

I.5 Restricciones relativas a la ubicación (informativo)

Se imponen muy pocas restricciones relativas a la ubicación de las casillas definidas en este anexo. Pueden ubicarse al final del fichero si se desea; probablemente, esta opción es conveniente cuando un fichero no indexado se indexa ulteriormente. Ahora bien, puede ser útil ubicar la casilla buscador de fichero hacia el principio del mismo, preferiblemente inmediatamente después de las casillas que obligatoriamente se han de agrupar contiguas al principio del fichero (por ejemplo, después de la casilla tipo de fichero en un fichero JP2 o después de la casilla requisitos del lector en un fichero JPX), una posición que resulta fácil de encontrar para los lectores del fichero. A fin de minimizar el desplazamiento de las casillas de fichero, cuando se añade esta casilla y, como opción, cuando se añade un código "jpip" a la lista de compatibilidad en la casilla tipo de fichero, podría utilizarse una casilla libre (definida en el anexo M.11.20 de la Rec. UIT-T T.801 | ISO/CEI 15444-2) como marcador de posición para este fin en un fichero pendiente de indexar.

Anexo J

Registro de ampliaciones a esta Recomendación | Norma Internacional

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

J.1 Introducción al registro

El registro es el procedimiento de añadir ampliaciones a esta Recomendación | Norma Internacional después de que se haya publicado. En esta Recomendación | Norma Internacional existen muchas capacidades que puedan ampliarse mediante registro. En esta subcláusula se indican los elementos que podrían ampliarse mediante registro, el proceso mediante el cual se pueden registrar estas capacidades y el proceso mediante el cual la Autoridad de Registro publica dichas ampliaciones. Solamente los elementos que se especifican en esta subcláusula pueden ampliarse mediante registro.

J.2 Elementos de registro

El proceso de registro se compone de los siguientes elementos:

- **Autoridad de registro:** La entidad organizativa encargada de revisar, mantener, distribuir y actuar como punto de contacto para todas las actividades relacionadas con el registro. La Autoridad de Registro se habrá de determinar.
- **Solicitante:** El solicitante es la persona u organización que solicita el registro de un elemento.
- **Junta de examen:** La Junta de examen es la entidad de organización que aprueba el registro de un elemento propuesto. Está formado por una Comisión ad hoc nombrada por el Presidente de la Junta de examen. La Junta de examen será el subgrupo JPIP de la ISO/CEI JTC 1/SC 29/WG 1.
- **Presidente de la Junta de examen:** El Presidente de la Junta de examen es el encargado de supervisar que se examine cada elemento presentado. El Presidente se pone en contacto con el solicitante a través de la Autoridad de Registro. El Presidente de la Junta de examen será el Presidente del subgrupo JPIP de la ISO/CEI JTC 1/SC 29/WG 1.
- **Prueba:** Examen que debe utilizar la Junta de examen para determinar si el elemento presentado debe registrarse.
- **Solicitud/elemento:** Este elemento es la propuesta de registro. Cada propuesta incluirá el nombre del elemento que desea ampliarse, la etiqueta/identidad propuesta para la ampliación y la razón/propósito de la ampliación.

J.3 Criterios de evaluación del registro

La Junta de examen evaluará todas las solicitudes basándose en los siguientes criterios:

- ¿La solicitud suple una necesidad que todavía no satisface la norma u otra ampliación?
- ¿La ampliación está lo suficientemente definida?
- ¿La ampliación suple una necesidad general (por ejemplo aplicaciones de emisión de secuencias de vídeo en general) o una necesidad específica de un fabricante (por ejemplo la implementación de la emisión de secuencias de vídeo de un determinado fabricante)?

J.4 Elementos que pueden ampliarse por registro

J.4.1 Casillas ampliadas dentro de una casilla marcador de posición

Podrán registrarse nuevos tipos de casilla para las casillas que se utilizarán dentro del campo `ExtendedBoxList` de una casilla marcador de ubicación (A.3.6.3). Toda propuesta para registrar un nuevo tipo de casilla deberá contener una definición completa de dicha casilla (tipo de casilla y contenido de la casilla), instrucciones relativas al momento en que el servidor puede escribir esta casilla dentro de la casilla marcador de posición, e instrucciones sobre qué debe hacer el cliente cuando encuentra una casilla marcador de posición que contiene esta casilla.

J.4.2 Contexto del tren de código

Podrán registrarse nuevos valores de `context-range` para solicitar trenes de código específicos que utilicen el campo contexto de tren de código (C.4.7). Toda propuesta para registrar un nuevo valor de `context-range` deberá contener una definición completa del valor, instrucciones sobre cómo el servidor debe hacer la correspondencia

de dicho valor con los trenes de código disponibles en el destino lógico e instrucciones sobre cómo el servidor debe responder en el encabezamiento de respuesta del contexto del tren de código.

J.4.3 Transporte de canal

Podrán registrarse nuevos transportes de canal (anexo H). La propuesta para registrar un nuevo transporte de canal deberá contener la definición completa del transporte, incluido el identificador que se utilizara para dicho transporte de canal.

J.4.4 Preferencias

Se podrán registrar nuevas preferencias de cliente. Esto incluye nuevos conjuntos de preferencias (nuevos valores de `related-pref-set` definidos en C.10.2.1) o nuevas opciones para el conjunto de preferencias existente o registrado. La propuesta para registrar una nueva opción de preferencia o un nuevo conjunto de preferencias contendrá la definición completa de la sintaxis, el significado de las nuevas opciones y las instrucciones de cómo debe responder el servidor para actuar de conformidad con dicha preferencia.

J.5 Proceso de registro

A continuación se describe el proceso de registro.

- a) El solicitante crea un elemento candidato para registro.
- b) Dicho elemento se transmite a la autoridad de registro.
- c) La utilidad de registro transmite el elemento al Presidente de la Junta de Examen.
- d) El Presidente de la Junta de Examen distribuye el elemento a la Junta de Examen y planifica reuniones, hace llamadas de teléfono, etc, según corresponda, para examinar el elemento.
- e) La Junta de Examen evalúa todas las solicitudes. Si el texto de una solicitud no cumple los requisitos, ésta se devolverá al solicitante para su aclaración. Se dará preferencia a las soluciones que son más generales, y las soluciones propuestas que sean muy específicas del fabricante podrán devolverse al solicitante para que las haga más generales y aplicables a toda la industria en general.
- f) En caso de aprobación, el Presidente transmite la aprobación a la Autoridad de Registro que notifica a la ISO y al solicitante, y pone a disposición el elemento registrado o publicado.
- g) En caso de denegación, el Presidente prepara un documento de respuesta que indica las razones por las que no se aprobó y lo transmite a la autoridad de registro, la cual, por su parte, notifica al solicitante.

J.6 Plazos para el proceso de registro

La Junta de Examen responderá a todas las solicitudes de registro en un plazo de siete meses desde la fecha de presentación. Dentro de ese periodo, la Junta de Examen se reunirá en una reunión oficial de la ISO/CEI JTC 1/SC 29/WG 1 para evaluar la propuesta, tomar una decisión y redactar la respuesta.

Anexo K

Ejemplos de aplicación

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

K.1 Introducción

En este anexo se dan ejemplos informativos de ciertos aspectos de las implementaciones JPIP.

K.2 Utilización del JPIP con trenes de código en otros formatos de fichero

El JPIP puede utilizarse para acceder trenes de código JPEG 2000 almacenados en un formato de fichero distinto de los formatos de la familia JPEG 2000, por ejemplo, los ficheros de DICOM y PDF tienen la capacidad de contener trenes de código JPEG 2000. En un entorno cliente-servidor, podrían utilizarse procedimientos no especificados en esta norma para localizar los trenes de código JPEG 2000. Una vez localizado el tren de código, podrían utilizarse peticiones de respuesta JPIP sobre dicho objeto. El campo petición de subdestino está pensado precisamente para situaciones de este tipo. Otra posibilidad sería que el servidor accediese a los trenes de código a través de un URL diferente.

K.3 Técnicas de implementación de la parte losa

K.3.1 Determinación por parte del servidor de las partes de losa pertinentes a una petición de una ventana seleccionada

Para comunicación mediante la parte losa, la correspondencia de una ventana seleccionada con un conjunto de losas es sencilla. La región deseada de la imagen se convierte a "unidades de la rejilla de referencia". Las porciones XT e YT del segmento marcado SIZ se utilizan para determinar qué losas forman parte de la ventana seleccionada.

NOTA – Aunque en la rejilla de referencia todas las losas tienen las mismas dimensiones, en una rejilla de referencia submuestreada, después de la descomposición en subbandas, no todas las losas tienen necesariamente las mismas dimensiones. La intersección de una losa con la ventana seleccionada, aun cuando la losa esté completamente contenida dentro de dicha ventana, puede tener como resultado que no haya ninguna muestra de la ventana seleccionada a los niveles más bajos de resolución; ahora bien, las implementaciones no necesitan sacar provecho de ello mediante la omisión de la losa totalmente de la respuesta.

El nivel de resolución y calidad se utilizan para determinar las partes de losas necesarias. La casilla tabla de índices de parte losa, si la hubiere, podría utilizarse para obtener información sobre la ubicación de las partes losa en el tren de códigos y (si se incluyen campos auxiliares) la finalización de niveles de resolución dentro de las partes losa. Los segmentos marcadores SOT también indican los índices de losa y de parte losa y el número de bytes de cada parte losa. Desde el tren de código, se transmiten al cliente los bytes adecuados, correspondientes a las partes losa que necesitan ser enviadas. En caso de que varíe la ventana seleccionada y, por tanto, las correspondientes losas, para actualizar la visualización de la imagen sólo habrán de enviarse las partes de losa respectivas que no hayan sido enviadas antes.

K.3.2 Decodificación de la imagen desde mensajes del tren JPT devuelto

El JPIP especifica mecanismos para comunicar datos y metadatos de la imagen comprimida entre el cliente y el servidor. Los mecanismos que dispone el cliente para visualizar los datos devueltos no están especificados, e incluso podría variar sobremedida según la aplicación. En esta subcláusula se proporciona información sobre la obtención de muestras de componente a partir de los datos devueltos.

La aplicación de cliente que ha recibido todos los datos del encabezamiento principal (lo que se indica por la aparición completa del bin de datos del encabezamiento en el mensaje de respuesta para el bin de datos de encabezamiento 0), puede concatenar ese bin de datos con todas las partes losa de los bins de datos de losa para formar un tren de código JPEG 2000 conforme. Este tren de códigos puede pasarse por un decodificador JPEG 2000 conforme y visualizar el resultado. Evidentemente, en aras de la eficiencia quizá el cliente estime conveniente proporcionar parámetros de la ventana seleccionada a un decodificador inteligente junto con el tren codificado a fin de visualizar únicamente las partes necesarias de la ventana seleccionada del caso.

K.3.3 Señalización auxiliar para partes losa

Los cuadros K.1 y K.2 ilustran la utilización de campos auxiliares en mensajes de bins de datos de losa ampliados y en la casilla tabla de índice de parte de losa.

NOTA – En este ejemplo, la definición de r varía de la utilizada en otros lugares de esta Recomendación | Norma Internacional, pero es compatible con el anexo B de la Rec. UIT-T T.800 | ISO/CEI 15444-1:2004.

El cuadro K.1 ilustra un caso sencillo en el que todas las componentes losa de una losa con resolución progresiva tienen el mismo número de niveles de descomposición y en el que las fronteras del mensaje (en el caso de bins de datos) o la frontera de parte de losa (en el caso de casilla de índice) sólo aparecen entre cada nivel de resolución sucesivo.

Cuadro K.1 – Ejemplo de utilización de los campos auxiliares en un caso sencillo

Número de secuencia de mensaje en el bin de datos o número de parte de losa en la losa	Nivel de resolución r	$n = N_L - r$	Valor auxiliar
0	0	2	2
1	1	1	1
2	2	0	0

El cuadro K.2 ilustra un caso más complicado en el que el número de niveles de descomposición varía en cada componente losa. En la última columna del cuadro se incluye un comentario sobre la primera vez que aparece un nuevo valor auxiliar. Este caso corresponde a una losa de una imagen de tres componentes, en un orden de progresión RC, por ejemplo el orden de progresión LRCP de una sola capa, o el orden de progresión RPCL con un solo recinto en la losa. Las fronteras del mensaje (en el caso de bin de datos) o las fronteras de la parte losa (en el caso de casilla de índice) aparecen una vez para cada componente de cada nivel de resolución así como entre niveles de resolución. Los componentes 0 y 1 tienen dos niveles de descomposición ($N_L = 2$) y el componente 2 tiene un solo nivel de descomposición ($N_L = 1$).

Cuadro K.2 – Ejemplo de utilización de campos auxiliares en un caso más complejo

Número de secuencia de mensaje en el bin de datos o número de parte de losa en la losa	Índice del componente c	Nivel de resolución r	$n = N_L - r$	Valor auxiliar	Comentario
0	0	0	2	3	Ningún nivel completo
1	1	0	2	2	$n = 2$ acaba de completarse
2	2	0	1	2	
3	0	1	1	2	
4	1	1	1	1	$n = 1$ también acaba de completarse
5	2	1	0	1	
6	0	2	0	1	
7	1	2	0	0	Todos los niveles se han completado

K.4 Técnicas de implementación basadas en recintos

K.4.1 Determinación por parte del servidor de los recintos relevantes de una petición de ventana seleccionada

Cuando la comunicación entraña un tipo de medios tren JPP, el servidor traduce la región de la imagen solicitada por el cliente en un conjunto de recintos que son pertinentes para la petición. La primera parte de este proceso implica la traducción de los parámetros fx , fy , sx , sy , ox y oy proporcionados por los campos de petición tamaño de cuadro, tamaño de región y traslación de región, en parámetros tamaño de cuadro, tamaño de región y traslación fx' , fy' , sx' , sy' , ox' y oy' para cada tren de código correspondiente. La traducción se lleva a cabo de la misma manera para los dos servicios, basados en recintos y basados en losas, utilizándose las ecuaciones C-1 y C-2, posiblemente modificadas de conformidad con las ecuaciones C-3 y C-4. En esta subcláusula se describe cómo debe determinar el servidor los recintos que son pertinentes para la región definida mediante los parámetros fx' , fy' , sx' , sy' , ox' y oy' dentro de un determinado tren de código.

Sea r un entero no negativo en la ecuación C-1 que utilizó el servidor para determinar fx' y fy' , basándose en la petición de cliente. Como se mencionó en relación con dicha ecuación, la forma más fácil de interpretar r es el número de niveles DWT de resolución más altos descartados, aún cuando r puede superar el número real de niveles DWT que hay disponibles para un determinado componente de losa. En primer lugar conviene hacer la correspondencia de la región descrita por sx' , sy' , ox' y oy' en la rejilla de alta resolución del tren de código. De este modo se obtiene una región cuya esquina superior izquierda viene dada por $(E_1^{\text{reg}}, E_2^{\text{reg}})$ y cuya esquina superior derecha viene dada por $(F_1^{\text{reg}} - 1, F_2^{\text{reg}} - 1)$, siendo:

$$E_1^{\text{reg}} = X\text{Osz} + 2^r \cdot ox', \quad E_2^{\text{reg}} = Y\text{Osz} + 2^r \cdot oy', \quad F_1^{\text{reg}} = E_1^{\text{reg}} + 2^r \cdot sx', \quad \text{y} \quad F_2^{\text{reg}} = E_2^{\text{reg}} + 2^r \cdot sy'$$

El servidor sólo tiene que tener en cuenta las losas cuya intersección con esta región en la rejilla de alta resolución del tren codificado es distinta de cero. Para cada una de estas losas, el servidor sólo necesita tener en cuenta los componentes de la imagen que fueran solicitados por el cliente, tal y como fueron descritos en relación con los campos de petición de contexto de componente y de tren de código. Para cada componente de losa considerado, indicado mediante t y c , sea, $D_{t,c}$ el número de niveles DWT que fueron utilizados para comprimir dicha componente de losa. Si $D_{t,c} \geq r$, el servidor debe descartar todos los recintos que pertenecen a los niveles de resolución más altos de r de la componente losa; de lo contrario, debe descartar todos los recintos que pertenecen a los niveles de resolución más altos $D_{t,c}$ de la componente losa, dejando únicamente los recintos que representan la subbanda LL más baja del componente de losa.

Para cada recinto que permanece después de descartar las losas, componentes y niveles de resolución mencionados anteriormente, el servidor debe identificar si los bloques de códigos que pertenecen a este recinto contribuyen a la reconstrucción de la región definida mediante $E_1^{\text{reg}}, E_2^{\text{reg}}$ y $F_1^{\text{reg}}, F_2^{\text{reg}}$ en la rejilla de alta resolución del tren de código. Un bloque de código contribuye a esta región si algunas de sus muestras afecta la reconstrucción de la muestra de imagen de plena resolución cuyas coordenadas (x,y) satisfacen:

$$E_1^{\text{reg}} \leq X\text{Rsz}^c \cdot x < F_1^{\text{reg}} \quad \text{y} \quad E_2^{\text{reg}} \leq Y\text{Rsz}^c \cdot y < F_2^{\text{reg}}$$

siendo $X\text{Rsz}^c$ y $Y\text{Rsz}^c$ los factores de submuestreo horizontales y verticales para la correspondiente componente, c , en el segmento marcador SIZ del tren de código.

Es importante tener presente que la reconstrucción de las componentes de la imagen de plena resolución implica la síntesis de ondícula, la cual es inherente al proceso de expansión. Por consiguiente, la región para la cual algún cierto recinto contribuye generalmente está superpuesta a las regiones para las cuales sus recintos adyacentes también contribuyen. El servidor debe prepararse para tener en cuenta estos efectos de expansión de la transformada ondícula cuando determine los recintos que son pertinentes para la petición del cliente.

En la sección 10.6.4 del libro "JPEG2000: image compression fundamentals, standards and practice" [11] se describe una forma de calcular las muestras de cualquier subbanda que contribuyen a una determinada región en la rejilla de alta resolución del tren de código. Para las regiones de subbanda, esto consiste simplemente en deducir los bloques de código que contribuyen y, por consiguiente, los recintos.

K.4.2 Decodificación de una imagen a partir de los mensajes de JPP devueltos

El JPIP especifica mecanismos para comunicar datos y metadatos de la imagen comprimida entre el cliente y el servidor. Los mecanismos de que dispone el cliente para visualizar los datos devueltos no se especifican y de hecho puede variar sobremanera según la aplicación.

K.5 Transcripción de protocolo JPIP

K.5.1 Introducción

En las transcripciones de ejemplo que figuran a continuación, los símbolos "<<<" al principio de la línea indican que el texto se envía del cliente al servidor, y los símbolos ">>>" al principio de la línea que se transmite del servidor al cliente; los símbolos "--" indican que el texto es un comentario y que en realidad no se transmite. Los comentarios también se utilizan para indicar que algunos de los datos transmitidos no se muestran en la transcripción.

K.5.2 Utilización de HTTP

La transcripción que figura a continuación muestra cinco peticiones que envía el cliente al servidor y la respuesta del servidor.

En la primera petición se pide el archivo JP2 denominado phoenix.jp2, se solicita el primer tren de código en el fichero, se incluye en la respuesta una longitud máxima, se solicita el id de destino, se solicita que los datos se devuelvan como un tren JPP y se solicita el establecimiento de una sesión por HTTP. No se solicita ninguna ventana, y por consiguiente tampoco datos de la imagen.

El servidor responde con el ID de destino para la imagen, y el ID para el nuevo canal establecido. La línea de encabezamiento que comienza con "JPIP-cnew" indica el nuevo trayecto que puede utilizarse para acceder al fichero imagen. El valor del trayecto "jpip" podría ser un trayecto hacia un programa CGI en el servidor designado para ocuparse de todas las instrucciones interactivas JPIP. Ciertos datos del fichero se devuelven en el cuerpo; éstos serán casillas del formato de fichero, y quizás el encabezamiento principal del primer tren de código.

La segunda petición del cliente utiliza el nuevo trayecto "jpip.cgi" y el ID del canal para identificar la imagen deseada (el nombre de la imagen o el ID de destino no son necesarios). En esta petición también se especifica una determinada ventana de interés.

La respuesta a la segunda petición indica que la ventana seleccionada se ha modificado y se devuelve una ventana más pequeña centrada en la ventana seleccionada solicitada. El servidor comienza a devolver los datos para dicha ventana.

Antes de recibir la respuesta completa a la segunda petición, el cliente realiza una tercera petición. El cliente ha ajustado su ventana seleccionada al tamaño especificado por el servidor.

A continuación, el servidor responde a la segunda petición durante un tiempo, y luego comienza a responder a la tercera petición. Durante esta respuesta, el cliente envía una cuarta petición con una región ligeramente diferente. El servidor continúa respondiendo a la tercera petición durante un tiempo y luego comienza a responder a la cuarta petición.

El cliente espera hasta haber recibido íntegramente la cuarta respuesta, y luego realiza una petición para terminar la sesión y la conexión HTTP. En este caso no hay datos de respuesta y la conexión se cierra.

```
<< GET /phoenix.jp2?stream=0&len=2000&tid=0&type=jpp-stream&cnew=http
HTTP/1.1
<< Host: dst-m
<<
>> HTTP/1.1 200 OK
>> JPIP-tid: 281B6E135135BBC0BC588452AC9B73C5
>> JPIP-cnew: cid=JPH_033C38BE48115AC9,path=jpip.cgi,transport=http
>> Cache-Control: no-cache
>> Transfer-Encoding: chunked
>> Content-Type: image/jpp-stream
>>
>> 102
-- 258 bytes of binary data
>> 0
>>
<< GET /jpip.cgi?fsiz=834,834&roff=0,0&rsiz=834,790&comps=0-
2&stream=0&len=2000&cid=JPH_033C38BE48115AC9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
>> HTTP/1.1 200 OK, with modifications
>> JPIP-roff: 120,114
>> JPIP-rsiz: 593,561
>> Cache-Control: no-cache
>> Transfer-Encoding: chunked
>> Content-Type: image/jpp-stream
>>
>> 393
-- 915 bytes of binary data
<< GET /jpip.cgi?fsiz=834,834&roff=120,114&rsiz=593,561&comps=0-
2&stream=0&len=2000&cid=JPH_033C38BE48115AC9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
>> 3f9
-- 1017 bytes of binary data
>> 0
>>
```

ISO/CEI 15444-9:2005 (S)

```
>> HTTP/1.1 200 OK
>> Cache-Control: no-cache
>> Transfer-Encoding: chunked
>> Content-Type: image/jpp-stream
>>
>> 359
  -- 857 bytes of binary data
<< GET /jpip.cgi?fsiz=834,834&roff=309,297&rsiz=121,86&comps=0-
2&stream=0&len=3906&cid=JPH_033C38BE48115AC9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
  >> 234
    -- 564 bytes of binary data
  >> 3d0
    -- 976 bytes of binary data
  >> 24f
    -- 591 bytes of binary data
  >> 0
  >>
  >> HTTP/1.1 200 OK
  >> Cache-Control: no-cache
  >> Transfer-Encoding: chunked
  >> Content-Type: image/jpp-stream
  >>
  >> 3b2
    -- 946 bytes of binary data
  >> 400
    -- 1024 bytes of binary data
  >> 263
    -- 611 bytes of binary data
  >> 356
    -- 854 bytes of binary data
  >> 209
    -- 521 bytes of binary data
  >> 0

<< GET /jpip.cgi?cclose=JPH_033C38BE48115AC9&len=0 HTTP/1.1
<< Host: dst-m
<< Connection: close
<< Cache-Control: no-cache
<<
```

A continuación se muestra un ejemplo de la instrucción GET HTTP basada en sesión con una petición de modelo.

```
<< GET /jpip.cgi?fsiz=1024,768&cid=JPH_5&model=Hm,H*,M*,P* HTTP/1.1
<< Host: jpip.jpeg.org
<< Cache-Control: no-cache

  >> HTTP/1.1 200 OK
  >> Cache-control: no-cache
  >> Transfer-Encoding: chunked
  >> 3
  -- 3 bytes of binary data
  >> 0
```

A continuación se muestra un ejemplo de la instrucción GET HTTP sin estados con una petición de modelo.

```
<< GET /images/kids.jp2?fsiz=1024,768&model=M0,Hm,H0:20,P0 HTTP/1.1
<< Host: jpip.jpeg.org
<< Cache-Control: no-cache

  >> HTTP/1.1 200 OK
  >> Cache-Control: no-cache
  >> Transfer-Encoding: chunked
```

```

>> Content-Type: image/jpp-stream
>> 400
-- 1024 bytes of binary data
>> 3f8
-- 1016 bytes of binary data
>> 0

```

K.5.3 Utilización de HTTP con respuesta TCP

```

<< GET /phoenix.jp2?stream=0&len=2000&tid=0&type=jpp-stream&cnew=http-
tcp,http HTTP/1.1
<< Host: dst-m
<<
  >> HTTP/1.1 200 OK
  >> JPIP-tid: 281B6E135135BBC0BC588452AC9B73C5
  >> JPIP-cnew: cid=JPHT033C38BE481154F9,path=jpip,transport=http-
tcp,auxport=80
  >> Cache-Control: no-cache
  >>
    << JPHT033C38BE481154F9 - [Note: This is the TCP channel connection
message]
    <<

<< GET /jpip.cgi?fsiz=834,834&roff=0,0&rsiz=834,790&comps=0-
2&stream=0&cid=JPHT033C38BE481154F9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
  >> HTTP/1.1 200 OK, with modifications
  >> JPIP-roff: 120,114
  >> JPIP-rsiz: 593,561
  >> Cache-Control: no-cache
  >>

<< GET /jpip.cgi?fsiz=834,834&roff=229,254&rsiz=155,113&comps=0-
2&stream=0&cid=JPHT033C38BE481154F9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
  >> HTTP/1.1 200 OK
  >> Cache-Control: no-cache
  >>

<< GET /jpip.cgi?fsiz=1667,1667&roff=457,507&rsiz=310,226&comps=0-
2&stream=0&cid=JPHT033C38BE481154F9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
  >> HTTP/1.1 200 OK
  >> Cache-Control: no-cache
  >>

<< GET /jpip.cgi?fsiz=3334,3334&roff=914,1014&rsiz=620,452&comps=0-
2&stream=0&cid=JPHT033C38BE481154F9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
  >> HTTP/1.1 200 OK
  >> Cache-Control: no-cache
  >>

```

ISO/CEI 15444-9:2005 (S)

```
<< GET /jpip.cgi?cclose=JPHT033C38BE481154F9 HTTP/1.1
<< Host: dst-m
<< Cache-Control: no-cache
<<
```

K.6 Utilización de JPIP con HTML

El sistema JPIP puede utilizarse en páginas HTML y XHTML de diversas formas. Si el servidor JPIP integra la capacidad de transcodificar porciones de una imagen al formato JPEG u otros tipos de medios de imagen completo, el HTML puede utilizarse para acceder a porciones de la imagen JPEG 2000 sin que sea necesario modificar los navegadores existentes.

Supóngase una página web que contiene el siguiente fragmento HTML:

```

```

Los navegadores web que desean visualizar esta página web con imágenes tienen que realizar una petición para obtener la imagen. Esta petición comenzará con:

```
GET /name.jp2?fsiz=128,128&rsiz=128,128&type=image/jpeg
Host: jpip.jpeg.org
```

e incluirá otras líneas de encabezamiento HTTP, que normalmente identifica al navegador, y los tipos de cosas que acepta el navegador. Esta petición HTTP es una petición JPIP aceptable y el servidor JPIP que reciba dicha petición podrá devolver un mensaje de error o bien determinar la porción correspondiente del fichero JP2 que se desea acceder y traducirlo al formato JPEG. El mensaje devuelto podría ser similar al siguiente:

```
HTTP/1.1 200 OK
Content-type: image/jpeg
Content-length: 20387
CRLF
JPEG-Compressed-Image-Data
```

Este mensaje es una respuesta JPIP aceptable, y además una respuesta HTTP aceptable que todos los navegadores de imágenes saben cómo visualizar. Obsérvese que es preferible, aunque no obligatorio, que el servidor utilice la codificación de transferencia fragmentada de modo que esta petición pueda interrumpirse. El ejemplo anterior no es un ejemplo de codificación de transferencia en fragmentos.

Es posible escribir páginas web que utilizarán JPEG solamente cuando el JPEG esté disponible, se utilice JPEG 2000 cuando esté disponible y se utilice trenes JPT o trenes JPP cuando estén disponibles en el navegador del cliente. Considérese el siguiente fragmento HTML:

```

```

En este caso no se solicita un tipo explícito. El servidor JPIP que utilice HTTP deberá por consiguiente examinar la línea "Accept:" de la petición HTTP expedida por el cliente. En función de la presencia de imagen/jp2 o imagen/tren jpt o imagen/tren jpp o imagen/jpeg, el servidor puede determinar el formato compatible que ha de devolver.

Anexo L

Colección ABNF del JPIP

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

L.1 ABNF de petición JPIP

```

;=====
; C.1.1 Request structure
;=====

jpip-request-field = target-field
                    / channel-field
                    / view-window-field
                    / metadata-field
                    / data-limit-field
                    / server-control-field
                    / cache-management-field
                    / upload-field
                    / client-cap-pref-field

target-field       = target           ; C.2.2
                    / subtarget       ; C.2.3
                    / tid              ; C.2.4

channel-field      = cid              ; C.3.2
                    / cnew            ; C.3.3
                    / cclose          ; C.3.4
                    / qid             ; C.3.5

view-window-field  = fsiz             ; C.4.2
                    / roff            ; C.4.3
                    / rsiz            ; C.4.4
                    / comps           ; C.4.5
                    / stream          ; C.4.6
                    / context         ; C.4.7
                    / srate           ; C.4.8
                    / roi             ; C.4.9
                    / layers          ; C.4.10

metadata-field     = metareq         ; C.5.2

data-limit-field   = len              ; C.6.1
                    / quality        ; C.6.2

server-control-field = align         ; C.7.1
                    / wait           ; C.7.2
                    / type           ; C.7.3
                    / drate          ; C.7.4

cache-management-field = model      ; C.8.1
                    / tpmodel        ; C.8.3
                    / need           ; C.8.4
                    / tpneed         ; C.8.5
                    / mset           ; C.8.6

upload-field       = upload          ; C.9.1

```

ISO/CEI 15444-9:2005 (S)

```
client-cap-pref-field = cap                ; C.10.1
                        / pref              ; C.10.2
                        / csf               ; C.10.3;
=====
; C.2.2 Target (target)
;=====
target = "target" "=" PATH

;=====
; C.2.3 Sub-target (subtarget)
;=====
subtarget = "subtarget" "=" byte-range
byte-range = UINT-RANGE

;=====
; C.2.4 Target ID (tid)
;=====
tid = "tid" "=" target-id
target-id = TOKEN

;=====
; C.3.1 Channel ID (cid)
;=====
cid = "cid" "=" channel-id
channel-id = TOKEN

;=====
; C.3.2 New Channel (cnew)
;=====
cnew = "cnew" "=" 1#transport-name
transport-name = TOKEN

;=====
; C.3.3 Channel Close (cclose)
;=====
cclose = "cclose" "=" ("*" / 1#channel-id)

;=====
; C.3.4 Request ID (qid)
;=====
qid = "qid" "=" UINT

;=====
; C.4.2 Frame Size (fsiz)
;=====
fsiz = "fsiz" "=" fx "," fy ["," round-direction]
fx = UINT
fy = UINT
round-direction = "round-up" / "round-down" / "closest"

;=====
; C.4.3 Offset (roff)
;=====
roff = "roff" "=" ox "," oy
ox = UINT
oy = UINT
```

```

;=====
; C.4.4 Region Size (rsiz)
;=====

rsiz = "rsiz" "=" sx "," sy
sx = UINT
sy = UINT

;=====
; C.4.5 Components (comps)
;=====

comps = "comps" "=" 1#UINT-RANGE

;=====
; C.4.6 Codestream (stream)
;=====

stream = "stream" "=" 1#sampled-range
sampled-range = UINT-RANGE [":" sampling-factor]
sampling-factor = UINT

;=====
; C.4.7 Codestream Context (context)
;=====

context = "context" "=" 1#context-range
context-range = jpxl-context-range / mj2t-context / reserved-context
jpxl-context-range = "jpxl" "<" jpx-layers ">" [ "[" jpxl-geometry "]" ]
jpx-layers = sampled-range
jpxl-geometry = "s" jpx-iset "i" jpx-inum
jpx-iset = UINT
jpx-inum = UINT
mj2t-context = "mj2t" "<" mj2-track ">" [ "[" mj2t-geometry "]" ]
mj2-track = NONZERO [ "+" "now" ]
mj2t-geometry = "track" / "movie"
reserved-context = 1*( TOKEN / "<" / ">" / "[" / "]" / "-" / ":" / "+" )

;=====
; C.4.8 Sampling Rate (srate)
;=====

srate = "srate" "=" streams-per-second
streams-per-second = UFLOAT

;=====
; C.4.9 ROI (roi)
;=====

roi = "roi" "=" region-name
region-name = 1*(DIGIT / ALPHA / "_")
              / "dynamic"

;=====
; C.4.10 Layers (layers)
;=====

layers = "layers" "=" UINT

```

ISO/CEI 15444-9:2005 (S)

```

;=====
; C.5.2 Metadata Request (metareq)
;=====
metareq = "metareq" "=" 1#("[ 1$(req-box-prop) "]" [root-bin] [max-depth])
          [metadata-only]
req-box-prop = box-type [limit] [metareq-qualifier] [priority]
limit = ":" (UINT / "r")
metareq-qualifier = "/" 1*("w" / "s" / "g" / "a")
priority = "!"
root-bin = "R" UINT
max-depth = "D" UINT
metadata-only = "!!"

;=====
; C.6.1 Maximum Response Length (len)
;=====
len = "len" "=" UINT

;=====
; C.6.2 Quality (quality)
;=====
quality = "quality" "=" (1*2DIGIT / "100")           ; 0 to 100

;=====
; C.7.1 Alignment (align)
;=====
align = "align" "=" ("yes" / "no")

;=====
; C.7.2 Wait (wait)
;=====
wait = "wait" "=" ("yes" / "no")

;=====
; C.7.3 Image Return Type (type)
;=====
type = "type" "=" 1#image-return-type
image-return-type = media-type / reserved-image-return-type
media-type = TOKEN "/" TOKEN *( ";" parameter )
reserved-image-return-type = TOKEN *( ";" parameter )
parameter = attribute "=" value
attribute = TOKEN
value = TOKEN

;=====
; C.7.4 Delivery Rate (drate)
;=====
drate = "drate" "=" rate-factor
rate-factor = UFLOAT
```

```

;=====
; C.8.1.1 Model (model)
;=====

model = "model" "=" 1#model-item
model-item = [codestream-qualifier ","] model-element
model-element = ["-"] bin-descriptor
bin-descriptor = explicit-bin-descriptor ; C.8.1.2
                / implicit-bin-descriptor ; C.8.1.3
codestream-qualifier = "[" 1$(codestream-range) "]"
codestream-range = first-codestream-id ["-"] [last-codestream-id]
first-codestream-id = UINT
last-codestream-id = UINT

;=====
; C.8.1.2 Explicit Form
;=====

explicit-bin-descriptor = explicit-bin
                        [":" (number-of-bytes / number-of-layers )]
explicit-bin = codestream-main-header-bin
              / meta-bin
              / tile-bin
              / tile-header-bin
              / precinct-bin

number-of-bytes = UINT
number-of-layers = %x4c UINT ; "L"
codestream-main-header-bin = %x48 %x6d ; "Hm"
meta-bin = %x4d bin-uid ; "M"
tile-bin = %x54 bin-uid ; "T"
tile-header-bin = %x48 bin-uid ; "H"
precinct-bin = %x50 bin-uid ; "P"
bin-uid = UINT / "*"

;=====
; C.8.1.3 Implicit Form
;=====

implicit-bin-descriptor = 1*implicit-bin [":" number-of-layers]
implicit-bin = implicit-bin-prefix (data-uid / index-range-spec)
implicit-bin-prefix = %x74 ; t -- tile
                    / %x63 ; c -- component
                    / %x72 ; r -- resolution level
                    / %x70 ; p -- position

index-range-spec = first-index-pos "-" last-index-pos
first-index-pos = UINT
last-index-pos = UINT
data-uid = UINT / "*"

```

```

;=====
; C.8.3 Tile-part Model involving JPT-streams (tpmodel)
;=====
tpmodel = "tpmodel" "=" 1#tpmodel-item
tpmodel-item = [codestream-qualifier "," ] tpmodel-element
tpmodel-element = ["-"] tp-descriptor
tp-descriptor = tp-range / tp-number
tp-range = tp-number "-" tp-number
tp-number = tile-number "." part-number
tile-number = UINT
part-number = UINT
;=====
; C.8.4 Need for Stateless Requests (need)
;=====
need = "need" "=" 1#need-item
need-item = [codestream-qualifier "," ] bin-descriptor
;=====
; C.8.5 Tile-part Need for Stateless Requests (tpneed)
;=====
tpneed = "tpneed" "=" 1#tpneed-item
tpneed-item = [codestream-qualifier "," ] tp-descriptor
;=====
; C.8.6 Model Set for Requests within a session (mset)
;=====
mset = "mset" "=" 1#sampled-range
;=====
; C.9.1 Upload (upload)
;=====
upload = "upload" "=" upload-type
upload-type = image-return-type ; C.7.3
;=====
; C.10.1 Client Capability (cap)
;=====
cap = "cap" "=" 1#capability-group
capability-group = processing-capability
                    / depth-capability
                    / config-capability
processing-capability = compatibility-capability
                    / vendor-capability
compatibility-capability = "cc." compatibility-code
vendor-capability = "vc." vendor-code [":" vendor-value]
vendor-code = 1*(LOWER / DIGIT / "." / "-")
vendor-value = TOKEN
depth-capability = "depth:" UINT
config-capability = "config:" UINT

```

```

;=====
; C.10.2.1 General
;=====

pref = "pref" "=" 1#(related-pref-set ["/r"])

related-pref-set = view-window-pref          ; C.10.2.2
                  / colour-meth-pref         ; C.10.2.3
                  / max-bandwidth            ; C.10.2.4
                  / bandwidth-slice          ; C.10.2.5
                  / placeholder-pref         ; C.10.2.6
                  / codestream-seq-pref     ; C.10.2.7
                  / other

other = TOKEN

;=====
; C.10.2.2 View-window handling preferences
;=====

view-window-pref = "fullwindow" / "progressive"

;=====
; C.10.2.3 Colour space method preference
;=====

color-meth-pref = 1$(color-meth [":" meth-limit])

color-meth = "color-enum" / "color-ricc" / "color-icc" / "color-vend"

meth-limit = UINT

;=====
; C.10.2.4 Max bandwidth
;=====

max-bandwidth = "mbw:" mbw

mbw = UINT ["K" / "M" / "G" / "T"]

;=====
; C.10.2.5 Bandwidth slice
;=====

bandwidth-slice = "slice:" slice

slice = NONZERO

;=====
; C.10.2.6 Placeholder preference
;=====

placeholder-pref = "meta:" placeholder-branch

placeholder-branch = "incr" / "equiv" / "orig"

;=====
; C.10.2.7 Codestream sequencing
;=====

codestream-seq-pref = "codeseq:" codestream-seq-option

codestream-seq-option = "sequential" / "reverse-sequential"
                      / "interleaved"

;=====
; C.10.3 Contrast sensitivity (csf)
;=====

csf = "csf" "=" 1#csf-sample-line

csf-sample-line = csf-density [":" csf-angle] ":" 1$sensitivity

csf-density = "density" ":" UFLOAT

```

csf-angle = "angle" ":" UFLOAT

sensitivity = UFLOAT

L.2 BNF de respuesta JPIP

```

;=====
; D.1.1 Reply structure
;=====

Status-Code = 3DIGIT

Reason-Phrase = *<TEXT, excluding CR and LF>

jpip-response-header =
    / JPIP-tid ; D.2.2
    / JPIP-cnew ; D.2.3
    / JPIP-qid ; D.2.4
    / JPIP-fsiz ; D.2.5
    / JPIP-rsiz ; D.2.6
    / JPIP-roff ; D.2.7
    / JPIP-comps ; D.2.8
    / JPIP-stream ; D.2.9
    / JPIP-context ; D.2.10
    / JPIP-roi ; D.2.11
    / JPIP-layers ; D.2.12
    / JPIP-srate ; D.2.13
    / JPIP-metareq ; D.2.14
    / JPIP-len ; D.2.15
    / JPIP-quality ; D.2.16
    / JPIP-type ; D.2.17
    / JPIP-mset ; D.2.18
    / JPIP-cap ; D.2.19
    / JPIP-pref ; D.2.20

;=====
; D.2.2 Target ID (JPIP-tid)
;=====

JPIP-tid = "JPIP-tid" ":" LWSP target-id

;=====
; D.2.3 New Channel (JPIP-cnew)
;=====

JPIP-cnew = "JPIP-cnew" ":" LWSP "cid" "=" channel-id
           ["," 1#(transport-param "=" TOKEN)]

transport-param = TOKEN

;=====
; D.2.4 Request ID (JPIP-qid)
;=====

JPIP-qid = "JPIP-qid" ":" LWSP UINT

;=====
; D.2.5 Frame Size (JPIP-fsiz)
;=====

JPIP-fsiz = "JPIP-fsiz" ":" LWSP fx "," fy

;=====
; D.2.6 Region Size (JPIP-rsiz)
;=====

JPIP-rsiz = "JPIP-rsiz" ":" LWSP sx "," sy

```



```

;=====
; D.2.7 Offset (JPIP-roff)
;=====

JPIP-roff = "JPIP-roff" ":" LWSP ox "," oy

;=====
; D.2.8 Components (JPIP-comps)
;=====

JPIP-comps = "JPIP-comps" ":" LWSP 1#UINT-RANGE

;=====
; D.2.9 Codestream (JPIP-stream)
;=====

JPIP-stream = "JPIP-stream" ":" LWSP 1#(prefixed-range / sampled-range)
prefixed-range = "<" ctxt-id ":" ctxt-elt ">" sampled-range
ctxt-id = UINT
ctxt-elt = UINT

;=====
; D.2.10 Codestream Context (JPIP-context)
;=====

JPIP-context = "JPIP-context" ":" LWSP 1$(context-range "=" 1#sampled-range)

;=====
; D.2.11 ROI (JPIP-roi)
;=====

JPIP-roi = "JPIP-roi" ":" LWSP
           "roi" "=" region-name ";"
           "fsiz" "=" UINT "," UINT ";"
           "rsiz" "=" UINT "," UINT ";"
           "roff" "=" UINT "," UINT ";"

region-name = 1*(DIGIT / ALPHA / "_")

;=====
; D.2.12 Layers (JPIP-layers)
;=====

JPIP-layers = "JPIP-layers" ":" LWSP UINT

;=====
; D.2.13 Sampling Rate (JPIP-srate)
;=====

JPIP-srate = "JPIP-srate" ":" LWSP UFLOAT

;=====
; D.2.14 Metadata request (JPIP-metareq)
;=====

JPIP-metareq = "JPIP-metareq" ":" LWSP
              1#( "[" 1$(req-box-prop) "]" [root-bin] [max-depth] )
              [metadata-only]

req-box-prop = box-type [limit] [metareq-qualifier] [priority]

;=====
; D.2.15 Maximum Response Length (JPIP-length)
;=====

JPIP-len = "JPIP-len" ":" LWSP UINT

;=====
; D.2.16 Quality (JPIP-quality)
;=====

```

ISO/CEI 15444-9:2005 (S)

```
JPIP-quality = "JPIP-quality" ":" LWSP (1*2DIGIT / "100" / "-1")
;=====
; D.2.17 Image Return Type (JPIP-type)
;=====
JPIP-type = "JPIP-type" ":" LWSP image-return-type
;=====
; D.2.18 Model Set (JPIP-mset)
;=====
JPIP-mset = "JPIP-mset" ":" LWSP 1#sampled-range
;=====
; D.2.19 Needed Capability (JPIP-cap)
;=====
JPIP-cap = "JPIP-cap" ":" LWSP 1#capability-code
;=====
; D.2.20 Unavailable Preference (JPIP-pref)
;=====
JPIP-pref = "JPIP-pref" ":" LWSP 1#related-pref-set
```

Anexo M

Declaración de patentes

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

La Organización Internacional de Normalización (ISO, *International Organization for Standardization*) y la Comisión Electrotécnica Internacional (CEI, *International Electrotechnical Commission*) desean poner de manifiesto que alegar la conformidad con esta parte de ISO/CEI 15444 puede implicar la utilización de patentes.

La ISO y la CEI se mantienen al margen en lo que respecta a la prueba, validez y alcance de estos derechos de patente.

El titular de estos derechos de patente ha garantizado a la ISO y a la CEI que están abiertos a la negociación de licencias de manera razonable y no discriminatoria para todo aquel que lo solicite desde cualquier parte del mundo. A este respecto, la declaración de los titulares de estos derechos de patente están registrados con la ISO y la CEI. Puede obtenerse información al respecto de las empresas que se enumeran a continuación.

Obsérvese que existe la posibilidad de que algunos elementos de esta parte de ISO/CEI 15444 podrían estar sujetos a derechos de patentes distintos de los indicados en este anexo. La ISO y la CEI no se hacen responsables en ningún caso de la indicación de algunos o todos los derechos de patente.

Empresa	
1	Canon Inc.
2	Ricoh Company, Limited.

Anexo N

Bibliografía

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

- [1] TAUBMAN (D.): Remote Browsing of JPEG 2000 Images, *Proc. Int. Conf. on Image Processing*, Vol. 1, pp. 229-232, septiembre de 2002.
- [2] LI (J.), SUN (H.), LI (H.), ZHANG (Q.), LIN (X.): Vfile – A Virtual File Media Access Mechanism and its Application in JPEG2000 Images for Browsing over Internet, *ISO/IEC JTC 1/SC 29/WG 1 Document Register: N1473*, noviembre de 1999.
- [3] BOLIEK (M.), WU (G.K.), GORMISH (M.J.): JPEG 2000 for Efficient Imaging in a Client/Server Environment, *Proc. SPIE Conf. on Applications of Digital Image Processing*, Vol. 4472, pp. 212-223, diciembre de 2001.
- [4] DESHPANDE (S.), ZENG (W.): Scalable Streaming of JPEG2000 Images Using Hypertext Transfer Protocol, *Proc. ACM Conf. on Multimedia*, pp. 372-381, octubre de 2001.
- [5] WRIGHT (A.), CLARK (R.), COLYER (G.): An Implementation of JPIP Based on HTTP, *ISO/IEC JTC 1/SC 29/WG 1 Document Register: N2426*, febrero de 2002.
- [6] GORMISH (M.), BANERJEE (S.): Tile-Based Transport of JPEG 2000, N. García, J.M. Martínez, L. Salgado (Eds.), VLVB03, LNCS 2849, pp. 217-224, 2003.
- [7] TAUBMAN (D.), ROSENBAUM (R.): Rate-Distortion Optimized Interactive Browsing of JPEG2000 Images, *Proc. Int. Conf. on Image Processing*, septiembre de 2003.
- [8] TAUBMAN (D.), PRANDOLINI (R.): Architecture, Philosophy and Performance of JPIP: Internet Protocol Standard for JPEG2000, presented at *Visual Communications and Image Processing*, Lugano, Suiza, 2003.
- [9] GORMISH (M.J.): TRUEW: Transport of Reversible and Unreversible EmbeddedWavelets (A JPIP Proposal), *ISO/IEC JTC 1/SC 29/WG 1 Document Register: N2602*, julio de 2002.
- [10] CANON: Proposal for JPIP Tier 2 protocol, *ISO/IEC JTC 1/SC 29/WG 1 Document Register: N2608*, junio de 2002.
- [11] TAUBMAN (D.), MARCELLIN (M.): JPEG2000: image compression fundamentals, standards and practice, *Kluwer Academic Publishers*, Boston, 2001.

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	Gestión de las telecomunicaciones, incluida la RGT y el mantenimiento de redes
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos, comunicaciones de sistemas abiertos y seguridad
Serie Y	Infraestructura mundial de la información, aspectos del protocolo Internet y Redes de la próxima generación
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación