



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**T.176**

(02/98)

SERIE T: TERMINALES PARA SERVICIOS DE  
TELEMÁTICA

---

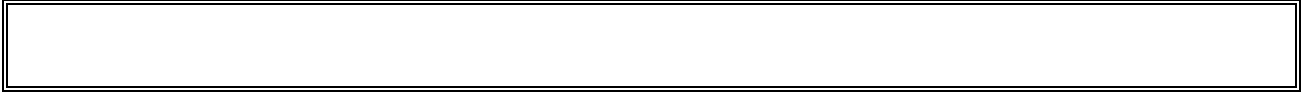
**Interfaz de programación de aplicaciones para  
instrucción y control de medios digitales de  
almacenamiento**

Recomendación UIT-T T.176

(Anteriormente Recomendación del CCITT)

---

RECOMENDACIONES DE LA SERIE T DEL UIT-T  
**TERMINALES PARA SERVICIOS DE TELEMÁTICA**



*Para más información, véase la Lista de Recomendaciones del UIT-T.*

## **RECOMENDACIÓN UIT-T T.176**

### **INTERFAZ DE PROGRAMACIÓN DE APLICACIONES PARA INSTRUCCIÓN Y CONTROL DE MEDIOS DIGITALES DE ALMACENAMIENTO**

#### **Resumen**

Esta Recomendación especifica la interfaz de programación de aplicaciones (API) de DSM-CC para uso en las aplicaciones multimedios básicas.

#### **Orígenes**

La Recomendación UIT-T T.176 ha sido preparada por la Comisión de Estudio 16 (1997-2000) del UIT-T y fue aprobada por el procedimiento de la Resolución N.º 1 de la CMNT el 6 de febrero de 1998.

## 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 Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), 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 N.º 1 de la CMNT.

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.

## 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 1998

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

## ÍNDICE

### Página

1	Alcance .....	1
2	Referencias normativas.....	1
3	Definiciones y abreviaturas.....	1
3.1	Definiciones .....	1
3.2	Abreviaturas.....	2
4	Visión general .....	2
4.1	Formato de intercambio de aplicación DAVIC .....	2
4.2	Núcleo de las API de Java .....	2
5	Paquete <code>org.apache.jackrabbit.davic.CosNaming</code> .....	4
5.1	Clase <code>org.apache.jackrabbit.davic.CosNaming.NameComponent</code> .....	4
5.2	Clase <code>org.apache.jackrabbit.davic.CosNaming.Binding</code> .....	4
5.3	Excepción <code>org.apache.jackrabbit.davic.CosNaming.NotFound</code> .....	4
5.4	Excepción <code>org.apache.jackrabbit.davic.CosNaming.CannotProceed</code> .....	4
5.5	Excepción <code>org.apache.jackrabbit.davic.CosNaming.InvalidName</code> .....	4
5.6	Clase <code>org.apache.jackrabbit.davic.CosNaming.BindingIterator</code> .....	5
5.7	Interfaz <code>org.apache.jackrabbit.davic.CosNaming.NamingContext</code> .....	5
6	Paquete <code>org.apache.jackrabbit.davic.dsmccuu</code> .....	5
6.1	Clase <code>org.apache.jackrabbit.davic.dsmccuu.Step</code> .....	5
6.2	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.SERVICE_XFR</code> .....	6
6.3	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.dsmccuuException</code> .....	6
6.4	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.INV_OFFSET</code> .....	6
6.5	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.INV_SIZE</code> .....	6
6.6	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.READ_LOCKED</code> .....	6
6.7	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.WRITE_LOCKED</code> .....	6
6.8	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.OPEN_LIMIT</code> .....	6
6.9	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.NO_AUTH</code> .....	7
6.10	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.UNK_USER</code> .....	7
6.11	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.BAD_COMPAT_INFO</code> .....	7
6.12	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.NO_RESUME</code> .....	7
6.13	Excepción <code>org.apache.jackrabbit.davic.dsmccuu.NO_SUSPEND</code> .....	7
6.14	Interfaz <code>org.apache.jackrabbit.davic.dsmccuu.Base</code> .....	7
6.15	Clase <code>org.apache.jackrabbit.davic.dsmccuu.File</code> .....	7
6.16	Clase <code>org.apache.jackrabbit.davic.dsmccuu.Directory</code> .....	8

	<b>Página</b>
6.17 Interfaz <code>davic.dsmccuu.SessionI</code> .....	9
6.18 Clase <code>davic.dsmccuu.Session</code> .....	9
6.19 Clase <code>davic.dsmccuu.SessionGateway</code> .....	10

## Recomendación T.176

# INTERFAZ DE PROGRAMACIÓN DE APLICACIONES PARA INSTRUCCIÓN Y CONTROL DE MEDIOS DIGITALES DE ALMACENAMIENTO

(Ginebra, 1998)

## 1 Alcance

La presente Recomendación especifica la interfaz de programación de aplicaciones (API, *application programming interface*) de DSM-CC para uso en las aplicaciones multimedios básicas. Esta Recomendación se aplica a los sistemas conformes a DAVIC.

## 2 Referencias normativas

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes.

- [1] ETS 300 777-3, *Terminal equipment (TE); End-to-end protocols for multimedia information retrieval services; Part 3: Application Programmable Interface (API) for MHEG-5.*
- [2] ISO/CEI DIS 13818-6, *Information technology – Generic coding of moving pictures and associated audio information – Part 6: Extension for digital storage media command and control.*
- [3] ISO/CEI 13522-5:1997, *Information technology – Coding of multimedia and hypermedia information - Part 5: Support for base-level interactive applications.*
- [4] ISO/CEI DIS 13522-6, *Information technology – Coding of multimedia and hypermedia information – Part 6: Support for enhanced interactive applications.*
- [5] ETS 300 777-1, *Terminal Equipment (TE); End-to-end protocols for multimedia information retrieval services; Part 1: Coding of multimedia and hypermedia information for basic multimedia applications (MHEG-5).*

## 3 Definiciones y abreviaturas

### 3.1 Definiciones

A los efectos de la presente Recomendación se aplica la definición de ISO/CEI DIS 13818-6 [2].

En esta Recomendación se definen los términos siguientes.

**3.1.1 interfaz de programación de aplicaciones (API):** Frontera a través de la cual una aplicación informatizada utiliza facilidades de lenguajes de programación para invocar servicios informatizados. Estas facilidades pueden incluir procedimientos u operaciones, objetos de datos compartidos y medios de resolución de identificadores.

**3.1.2 aplicación local:** Parte de un soporte lógico que forma parte de la aplicación (de telecomunicación) y funciona en el equipo considerado.

## 3.2 Abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

API	Interfaz de programación de aplicación ( <i>application programming interface</i> )
ASN.1	Notación de sintaxis abstracta uno ( <i>abstract syntax notation one</i> )
DAVIC	Consejo audiovisual digital ( <i>digital audio visual council</i> )
DSM-CC	Instrucción y control de medios de almacenamiento digital ( <i>digital storage media command and control</i> )
MHEG	Grupo de expertos en codificación de información multimedios e hipermedios ( <i>multimedia and hypermedia information coding experts group</i> )
SI	Información de servicio ( <i>service information</i> )
STU	Unidad superior ( <i>set top unit</i> )
VM	Máquina virtual ( <i>virtual machine</i> )

## 4 Visión general

La cláusula siguiente sitúa la API definida por esta Recomendación en el marco de las especificaciones DAVIC.

### 4.1 Formato de intercambio de aplicación DAVIC

Para entregar información multimedios a las STU de manera interoperable, las aplicaciones utilizarán el formato de intercambio de forma final MHEG-5, definido por ISO/CEI 13522-5 [3]. Se utilizará la codificación y la notación ASN.1 definidas por ETS 300 777-1 [4] para intercambiar objetos MHEG-5. Este formato define la semántica y la codificación de los objetos multimedios e hipermedios.

Para entregar códigos programa a las STU de manera interoperable, las aplicaciones utilizarán la clase programa intercambiado (*InterchangedProgram*) MHEG-5 para encapsular el código VM Java<sup>1</sup>, de conformidad con la semántica y la codificación definidas por ISO/CEI DIS 13522-6 [5]. Las clases de VM Java son llamadas desde objetos MHEG-5, utilizando las acciones elementales llamada (*call*) y diapasón (*fork*) MHEG-5.

La unidad de intercambio de código VM Java es una clase de VM Java. Las clases de VM Java se codificarán como se define en la sección formato de fichero de clase de la especificación de máquina virtual Java. Una clase Java encapsula datos y métodos que consisten en secuencias de instrucciones. El conjunto de instrucciones viene definido por la sección conjunto de instrucciones de máquina virtual Java de la especificación de máquina virtual Java.

### 4.2 Núcleo de las API de Java

El código VM Java utiliza el siguiente conjunto de API en las especificaciones DAVIC 1.1 [1] para expresar acceso a funciones básicas de la STU de manera interoperable:

- el paquete `java.lang`;

---

<sup>1</sup> Java es una marca comercial o una marca registrada de Sun Microsystems, Inc.



- el paquete `java.util`;
- el paquete `iso.mheg5`;
- el paquete `davic.dsmccuu`;
- el paquete `etsi.si`.

NOTA 1 – La especificación VM Java proporciona mecanismos flexibles para invocar funciones externas cuya interfaz está definida como un paquete Java. La especificación DAVIC 1.1 incluye únicamente un núcleo mínimo de paquetes necesarios para que el código VM Java sea útil en un entorno DAVIC. Se prevé que ulteriormente se normalizarán paquetes Java adicionales.

NOTA 2 – Especialmente, el paquete `java.io`, si bien no es estrictamente necesario para el funcionamiento útil del entorno VM, forma parte de las clases básicas de Java. Se prevé añadir el paquete `java.io` al núcleo DAVIC de API Java junto con una especificación adecuada de su semántica en un entorno DAVIC.

El paquete `java.lang`, definido por la documentación API de Java, consiste en el conjunto mínimo de clases VM Java que se requieren para utilizar el código VM Java, que admite la funcionalidad siguiente: tipos de datos básicos, objeto, operaciones matemáticas, seguridad, gestión de hilos, manipulación de cadenas, tratamiento de excepciones.

El paquete `java.util`, definido por la documentación API de Java, consiste en las clases VM de Java que admiten varias características de utilidades comunes a todos los programas VM Java.

El paquete `iso.mheg5`, definido por ETS 300 777-3 [1], proporciona un código VM Java con acceso a objetos de interacción y presentación multimedios MHEG-5 y manipulación de los mismos, es decir acceso a los atributos dinámicos de los objetos MHEG-5 e invocación de acciones elementales sobre objetos MHEG-5.

El paquete `davic.dsmccuu` y el paquete asociado `davic.CosNaming` permiten que el código VM Java utilice los objetos de interfaz U-U DSM-CC para acceso a los datos de red.

El paquete `davic.dsmccuu` y el paquete asociado `davic.CosNaming` dan acceso a un subconjunto de las API U-U DSM-CC definido por ISO/CEI DIS 13818-6. Este subconjunto consiste en:

- las operaciones `listar (list)` y `resolver (resolve)` de la interfaz abstracta contexto de denominación (`NamingContext`);
- las operaciones `cerrar (close)` y `destruir (destroy)` de la interfaz abstracta base (`Base`);
- las operaciones `siguiente (next_one)` y `siguientes n (next_n)` de la interfaz instanciable iterador vinculante (`BindingIterator`);
- las operaciones `abrir (open)` y `cerrar (close)` de la interfaz instanciable directorio (`Directory`);
- las operaciones `leer (read)` y `escribir (write)` así como el atributo de sólo lectura tamaño de contenido (`ContentSize`) de la interfaz instanciable fichero (`File`);
- las operaciones `unir (attach)` y `desprender (detach)` de la interfaz instanciable sesión (`Session`);
- la interfaz instanciable cabecera de sesión (`SessionGateway`).

Gracias al paquete `etsi.si`, el código VM Java puede acceder a la información transmitida en el tren de información de servicio (SI, *service information*) DAVIC.

## 5 Paquete `davic.CosNaming`

### 5.1 Clase `davic.CosNaming.NameComponent`

```
package davic.CosNaming;  
  
public class NameComponent {  
    public String id;  
    public String kind;  
}
```

### 5.2 Clase `davic.CosNaming.Binding`

```
package davic.CosNaming;  
  
public class Binding {  
    // constant declarations for the "binding_type" attribute  
    public static final short nobject = 0;  
    public static final short ncontext = 1;  
  
    public NameComponent[] binding_name;  
    public int binding_type;  
}
```

### 5.3 Excepción `davic.CosNaming.NotFound`

```
package davic.CosNaming;  
  
public class NotFound extends Exception{  
    // constant declarations for the "why" attribute  
    public static final short missing_node = 0;  
    public static final short not_context = 1;  
    public static final short not_object = 2;  
  
    public int why;  
    public NameComponent[] rest_of_name;  
}
```

### 5.4 Excepción `davic.CosNaming.CannotProceed`

```
package davic.CosNaming;  
  
public class CannotProceed extends Exception{  
    public NamingContext ext;  
    public NameComponent[] rest_of_name;  
}
```

### 5.5 Excepción `davic.CosNaming.InvalidName`

```
package davic.CosNaming;  
  
public class InvalidName extends Exception{  
}
```

## 5.6 Clase `davic.CosNaming.BindingIterator`

```
package davic.CosNaming;

public class BindingIterator {
    public boolean next_one(
        Binding b
    )
    {
        // actual code shall be inserted here
        return true;
    }

    public void next_n(
        int how_many,
        Binding[] bl
    )
    {
        // actual code shall be inserted here
    }

    public void destroy()
    {
        // actual code shall be inserted here
    }
}
```

## 5.7 Interfaz `davic.CosNaming.NamingContext`

```
package davic.CosNaming;

public interface NamingContext {
    public void list(
        int how_many,
        Binding[] bl,
        BindingIterator bi
    );

    public Object resolve(
        NameComponent[] n
    )throws NotFound, CannotProceed, InvalidName;
}
```

## 6 Paquete `davic.dsmccuu`

### 6.1 Clase `davic.dsmccuu.Step`

```
package davic.dsmccuu;

import davic.CosNaming.*;

public class Step {
    public NameComponent name;
    public boolean process;
}
```

## 6.2 Excepción `davic.dsmccuu.SERVICE_XFR`

```
package davic.dsmccuu;

import davic.CosNaming.*;

public class SERVICE_XFR extends Exception {
    // service location
    public byte[] serviceDomain;
    public NameComponent[] pathName;
    public byte[] initialContext;
}
```

## 6.3 Excepción `davic.dsmccuu.dsmccuuException`

```
package davic.dsmccuu;

public class dsmccuuException extends Exception {
    public short minor;
    public short completed;
}
```

## 6.4 Excepción `davic.dsmccuu.INV_OFFSET`

```
package davic.dsmccuu;

public class INV_OFFSET extends dsmccuuException {
}
```

## 6.5 Excepción `davic.dsmccuu.INV_SIZE`

```
package davic.dsmccuu;

public class INV_SIZE extends dsmccuuException {
}
```

## 6.6 Excepción `davic.dsmccuu.READ_LOCKED`

```
package davic.dsmccuu;

public class READ_LOCKED extends dsmccuuException {
}
```

## 6.7 Excepción `davic.dsmccuu.WRITE_LOCKED`

```
package davic.dsmccuu;

public class WRITE_LOCKED extends dsmccuuException {
}
```

## 6.8 Excepción `davic.dsmccuu.OPEN_LIMIT`

```
package davic.dsmccuu;

public class OPEN_LIMIT extends dsmccuuException {
}
```

## 6.9 Excepción `davic.dsmccuu.NO_AUTH`

```
package davic.dsmccuu;  
  
public class NO_AUTH extends dsmccuuException {  
    public byte[] authData;  
}
```

## 6.10 Excepción `davic.dsmccuu.UNK_USER`

```
package davic.dsmccuu;  
  
public class UNK_USER extends dsmccuuException {  
}
```

## 6.11 Excepción `davic.dsmccuu.BAD_COMPAT_INFO`

```
package davic.dsmccuu;  
  
public class BAD_COMPAT_INFO extends dsmccuuException {  
}
```

## 6.12 Excepción `davic.dsmccuu.NO_RESUME`

```
package davic.dsmccuu;  
  
public class NO_RESUME extends dsmccuuException {  
}
```

## 6.13 Excepción `davic.dsmccuu.NO_SUSPEND`

```
package davic.dsmccuu;  
  
public class NO_SUSPEND extends dsmccuuException {  
}
```

## 6.14 Interfaz `davic.dsmccuu.Base`

```
package davic.dsmccuu;  
  
interface Base {  
    public void close();  
  
    public void destroy();  
}
```

## 6.15 Clase `davic.dsmccuu.File`

```
package davic.dsmccuu;  
  
public class File implements Base {  
    // Base.close implementation  
    public void close()  
    {  
        // actual code shall be inserted here  
    }  
}
```

```

// Base.destroy implementation
public void destroy()
{
// actual code shall be inserted here
}

public int[] getContentSize()
{
// actual code shall be inserted here
return null;
}

public void read(
    int[] aOffset,
    int aSize,
    boolean aReliable,
    byte[] rData
) throws
    INV_OFFSET, INV_SIZE, READ_LOCKED
{
// actual code shall be inserted here
}

public void write(
    int[] aOffset,
    int aSize,
    byte[] rData
) throws
    INV_OFFSET, INV_SIZE, WRITE_LOCKED
{
// actual code shall be inserted here
}
}

```

## 6.16 Clase `davic.dsmccuu.Directory`

```

package davic.dsmccuu;

import davic.CosNaming.*;

public class Directory implements NamingContext {
    // NamingContext.list implementation
    public void list(
        int how_many,
        Binding[] bl,
        BindingIterator bi
    )
    {
// actual code shall be inserted here
}

    // NamingContext.resolve implementation
    public Object resolve(
        NameComponent[] n
    )throws NotFound, CannotProceed, InvalidName
    {
// actual code shall be inserted here
return null;
}
}

```

```

public void open(
    char aPathType,
    Step[] rPathStep,
    Object[] resolvedRefs
) throws
    OPEN_LIMIT, NO_AUTH, UNK_USER, SERVICE_XFR,
    NotFound, CannotProceed, InvalidName
{
    // actual code shall be inserted here
}

public void close()
{
    // actual code shall be inserted here
}
}

```

## 6.17 Interfaz `davic.dsmccuu.SessionI`

```

package davic.dsmccuu;

import davic.CosNaming.*;

interface SessionI {
    public void attach(
        byte[] serviceDomain,
        NameComponent[] pathName,
        byte[] userContext,
        Object[] resolvedRefs
    ) throws
        OPEN_LIMIT, NO_AUTH, UNK_USER, SERVICE_XFR, BAD_COMPAT_INFO, NO_RESUME,
        NotFound, CannotProceed, InvalidName;

    public void detach(
        boolean aSuspend,
        byte[] savedContext
    ) throws
        NO_SUSPEND;
}

```

## 6.18 Clase `davic.dsmccuu.Session`

```

package davic.dsmccuu;

public class Session implements SessionI {
    // SessionI.attach implementation
    public void attach(
        byte[] serviceDomain,
        NameComponent[] pathName,
        byte[] userContext,
        Object[] resolvedRefs
    ) throws
        OPEN_LIMIT, NO_AUTH, UNK_USER, SERVICE_XFR, BAD_COMPAT_INFO, NO_RESUME,
        NotFound, CannotProceed, InvalidName
    {
        // actual code shall be inserted here
    }
}

```

```

// SessionI.detach implementation
public void detach(
    boolean aSuspend,
    byte[] savedContext
) throws
    NO_SUSPEND
{
// actual code shall be inserted here
}
}

```

## 6.19 Clase `davic.dsmccuu.SessionGateway`

```
package davic.dsmccuu;
```

```

public class SessionGateway extends Directory implements SessionI {
// SessionI.attach implementation
public void attach(
    byte[] serviceDomain,
    NameComponent[] pathName,
    byte[] userContext,
    Object[] resolvedRefs
) throws
    OPEN_LIMIT, NO_AUTH, UNK_USER, SERVICE_XFR, BAD_COMPAT_INFO, NO_RESUME,
    NotFound, CannotProceed, InvalidName
{
// actual code shall be inserted here
}

// SessionI.detach implementation
public void detach(
    boolean aSuspend,
    byte[] savedContext
) throws
    NO_SUSPEND
{
// actual code shall be inserted here
}
}

```



## SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
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	Transmisiones de señales radiofónicas, de televisión 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	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
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
<b>Serie T</b>	<b>Terminales para servicios de telemática</b>
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información
Serie Z	Lenguajes de programación