



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**X.780**

**Corrigendum 1**  
(10/2001)

SERIE X: REDES DE DATOS Y COMUNICACIÓN  
ENTRE SISTEMAS ABIERTOS

Gestión de interconexión de sistemas abiertos –  
Funciones de gestión y funciones de arquitectura de  
gestión distribuida abierta

---

Directrices de la RGT para la definición de objetos  
gestionados mediante arquitectura de intermediario  
de petición de objeto común

**Corrigendum 1**

Recomendación UIT-T X.780 (2001) – Corrigendum 1

(Anteriormente Recomendación del CCITT)

---

RECOMENDACIONES UIT-T DE LA SERIE X  
REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

<b>REDES PÚBLICAS DE DATOS</b>	
Servicios y facilidades	X.1–X.19
Interfaces	X.20–X.49
Transmisión, señalización y conmutación	X.50–X.89
Aspectos de redes	X.90–X.149
Mantenimiento	X.150–X.179
Disposiciones administrativas	X.180–X.199
<b>INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Modelo y notación	X.200–X.209
Definiciones de los servicios	X.210–X.219
Especificaciones de los protocolos en modo conexión	X.220–X.229
Especificaciones de los protocolos en modo sin conexión	X.230–X.239
Formularios para declaraciones de conformidad de implementación de protocolo	X.240–X.259
Identificación de protocolos	X.260–X.269
Protocolos de seguridad	X.270–X.279
Objetos gestionados de capa	X.280–X.289
Pruebas de conformidad	X.290–X.299
<b>INTERFUNCIONAMIENTO ENTRE REDES</b>	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.369
Redes basadas en el protocolo Internet	X.370–X.399
<b>SISTEMAS DE TRATAMIENTO DE MENSAJES</b>	X.400–X.499
<b>DIRECTORIO</b>	X.500–X.599
<b>GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS</b>	
Gestión de redes	X.600–X.629
Eficacia	X.630–X.639
Calidad de servicio	X.640–X.649
Denominación, direccionamiento y registro	X.650–X.679
Notación de sintaxis abstracta uno	X.680–X.699
<b>GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Marco y arquitectura de la gestión de sistemas	X.700–X.709
Servicio y protocolo de comunicación de gestión	X.710–X.719
Estructura de la información de gestión	X.720–X.729
<b>Funciones de gestión y funciones de arquitectura de gestión distribuida abierta</b>	<b>X.730–X.799</b>
<b>SEGURIDAD</b>	X.800–X.849
<b>APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899
<b>PROCESAMIENTO DISTRIBUIDO ABIERTO</b>	X.900–X.999

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

## **Recomendación UIT-T X.780**

### **Directrices de la RGT para la definición de objetos gestionados mediante arquitectura de intermediario de petición de objeto común**

#### **CORRIGENDUM 1**

#### **Orígenes**

El corrigendum 1 a la Recomendación UIT-T X.780 (2001), preparado por la Comisión de Estudio 4 (2001-2004) del UIT-T, fue aprobado por el procedimiento de la Resolución 1 de la AMNT el 7 de octubre de 2001.

## 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.

## 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 2002

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

	<b>Página</b>
1) Subcláusula 2.1 .....	1
2) Subcláusula 5.1.3 .....	1
3) Subcláusula 5.5 .....	1
4) Subcláusula 6.3 .....	1
5) Nueva subcláusula 6.3.6 .....	1
6.3.6 Tipos comunes .....	2
6) Subcláusula 6.4 .....	2
7) Subcláusula 6.6 .....	2
8) Subcláusula 6.9.1 .....	2
6.9.1 Operaciones de creación .....	2
9) Subcláusula 6.9.1.2 .....	3
10) Nueva subcláusula 6.9.1.3 .....	3
11) Subcláusula 6.9.1.4 .....	3
12) Subcláusula 9.1 .....	3
13) Anexo A .....	3
14) Nuevo Anexo C .....	5
Anexo C – El módulo IDL CORBA para tipos comunes .....	5



## Recomendación UIT-T X.780

### Directrices de la RGT para la definición de objetos gestionados mediante arquitectura de intermediario de petición de objeto común

#### 1) **Subcláusula 2.1**

*Añádase una nueva referencia:*

- [8] UIT-T X.739 (1993) | ISO/CEI 10164-11 (1994), *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Objetos métricos y atributos.*

#### 2) **Subcláusula 5.1.3**

*Después de la segunda oración del primer párrafo añádase la siguiente oración:*

Se definen constantes de cadena para cada nombre de cadena de lote condicional.

#### 3) **Subcláusula 5.5**

*En el IDL, suprimanse las siguientes líneas.*

```
valuetype CreateErrorInfoType : ApplicationErrorInfoType {
    public MOSetType      relatedObjects;
    public AttributeSetType attributeList;
};

valuetype DeleteErrorInfoType : ApplicationErrorInfoType {
    public MOSetType      relatedObjects;
    public AttributeSetType attributeList;
};
```

*y sustitúyanse por las siguientes:*

```
valuetype CreateErrorInfoType : ApplicationErrorInfoType {
    public MONameSetType  relatedObjects;
    public AttributeSetType attributeList;
};

valuetype DeleteErrorInfoType : ApplicationErrorInfoType {
    public MONameSetType  relatedObjects;
    public AttributeSetType attributeList;
};
```

#### 4) **Subcláusula 6.3**

*Suprimase la última línea del primer párrafo:*

Futuras versiones de CORBA planean permitir excepciones definidas por el usuario en los accesos a los atributos, y estas directrices pueden cambiar para aprovecharlo.

*Después del primer párrafo, añádase el siguiente:*

Deberá evitarse la definición de atributos que contengan referencias a objetos gestionados. En su lugar se deben utilizar nombres de objetos gestionados.

#### 5) **Nueva subcláusula 6.3.6**

*Añádase la siguiente nueva subcláusula 6.3.6 con el título "Tipos comunes":*

### 6.3.6 Tipos comunes

El anexo C contiene un módulo de definición de tipos comunes. Estos tipos son utilizados en más de una Recomendación. Este módulo se utiliza para proporcionar un lugar único para la definición de estos tipos. El módulo será completado con otros tipos cuando sea necesario.

#### 6) *Subcláusula 6.4*

*En el primer párrafo, después de la segunda oración, añádase lo siguiente:*

Sin embargo, se deberá evitar la utilización de parámetros que contengan referencia a objetos gestionados. En su lugar se deben utilizar nombres de objetos gestionados.

#### 7) *Subcláusula 6.6*

*Después del tercer párrafo, añádase el siguiente:*

Además, para cada lote condicional definido, se definirá una cadena constante que se podrá utilizar en el atributo de lotes para ejemplares de objeto gestionado que utilicen ese lote condicional. Es importante calificar completamente esta cadena para distinguir los lotes de diferentes ficheros IDL, por ejemplo:

```
const string administrativeStatePackage =  
    "itut_m3120::administrativeStatePackage";
```

#### 8) *Subcláusula 6.9.1*

*Sustitúyase la subcláusula 6.9.1 por la siguiente:*

### 6.9.1 Operaciones de creación

Cada interfaz factoría definirá una única operación que los clientes utilizarán para crear objetos. El nombre de esta operación será "create" (creación) y devolverá una referencia a *ManagedObject*. Ésta se utilizará para devolver una referencia al objeto de nueva creación, que podrá entonces ser "estrechado" por el cliente al tipo de interfaz específico para ese objeto, o utilizado directamente para operaciones de clase base. Los cinco primeros parámetros de cada operación creación son siempre los mismos. Después de ellos vienen parámetros para cada atributo escribible o asignable en creación definido para el objeto gestionado. (Un atributo asignable en creación, es uno para el que el objeto no tiene operación "set", pero para el que está especificado un valor durante la operación creación.) Los nombres de estos parámetros son iguales al nombre del atributo. (Éste es el nombre de una operación de acceso al atributo sin el "Get" o "Set" final.) Cada operación de creación tiene también que aceptar parámetros para asignar valores de cualquier atributo que sea escribible o asignable en creación de todas las superclases del objeto creado por la factoría. A continuación se presenta un ejemplo de una operación de creación para una factoría de equipo:

```
ManagedObject create(  
    in NameBindingType nameBinding,           // module name containing Name Binding  
                                           // info.  
    in MONameType superiorObject,           // Name of containing object.  
    in string reqID,                         // Requested ID value for name, will be  
                                           // empty if auto-naming is to be used.  
    out MONameType name,                    // Entire name of newly created object.  
    in StringSetType packages,              // List of packages requested.  
    ...                                       // Writeable and set-by-create values  
                                           // for Equipment superclass attributes.  
    ...                                       // Writeable and set-by-create values  
                                           // for Equipment attributes.  
);
```

**9) Subcláusula 6.9.1.2**

*En la primera oración del primer párrafo, sustitúyanse las palabras "es una referencia al" por "es el nombre del". En la tercera oración del primer párrafo, sustitúyanse las palabras "la referencia suministrada" por "el objeto superior". En la última oración del segundo párrafo, sustitúyanse las palabras "la referencia del objeto superior será nula" por "el nombre será una cadena vacía".*

**10) Nueva subcláusula 6.9.1.3**

*Insértese una nueva subcláusula 6.9.1.3 con el título "Identificador solicitado" y modifíquese la numeración de la actual subcláusula 6.9.1.3 y las siguientes.*

**6.9.1.3 Identificador solicitado**

El tercer parámetro es el identificador que se ha de asignar al nuevo objeto según la solicitud. Esta cadena será el campo *ID* de la vinculación de nombre CORBA creada en el servicio de denominación CORBA para el nuevo objeto. Por tanto, esta cadena *ID* y el campo *kind* contenido en el módulo de vinculación de nombres, es una denominación del nuevo objeto en relación con el objeto superior. Si el cliente suministra una cadena vacía para este parámetro, la factoría elegirá automáticamente un valor de *ID*. Si el valor de *ID* proporcionado es un duplicado, la factoría formula una excepción *duplicateName CreateError*. (Esto significa que tanto el campo *ID* como el campo *kind* concuerdan con un objeto existente contenido por el objeto superior.)

**11) Subcláusula 6.9.1.4**

*Sustitúyase el contenido de la subcláusula 6.9.1.4 (según la nueva numeración del punto anterior, y que tiene por título "Nombre") por lo siguiente:*

La factoría utilizará el cuarto parámetro para retornar al cliente el nombre completo del objeto gestionado recién creado.

**12) Subcláusula 9.1**

*Después del inciso 8 de la enumeración, añádase el siguiente:*

9) Cumplir los convenios para las constantes de cadena de lote condicional especificados en 6.6.

*Se debe adaptar la numeración de los apartados siguientes.*

**13) Anexo A**

*En el IDL, suprimanse las siguientes líneas:*

```
/** Un SuspectObject identifica un objeto sospechoso que puede ser la causa de un
fallo. Normalmente es un componente de una SuspectObjectList.
@member objectClass          Object class of the suspect object
@member suspectObjectInstance Object instance of the suspect
                             object
@member failureProbability   Optional failure responsibility
                             probability from 1 to 100
*/

struct SuspectObjectType {
    ObjectClassType          objectClass;
    ManagedObject            suspectObjectInstance;
    UnsignedShortTypeOpt    failureProbability;
};
```

*y sustitúyanse por las siguientes:*

```
/** Un SuspectObject identifica un objeto que puede ser la causa de un fallo.
Normalmente es un componente de una SuspectObjectList.
@member objectClass      Object class of the suspect object
@member suspectObject    Name of the suspect object
@member failureProbability Optional failure responsibility
                          probability from 1 to 100
*/

struct SuspectObjectType {
    ObjectClassType      objectClass;
    MONameType           suspectObject;
    UnsignedShortTypeOpt failureProbability;
};
```

*Suprímense las siguientes líneas:*

```
valuetype CreateErrorInfoType : ApplicationErrorInfoType {
    public MOSetType      relatedObjects;
    public AttributeSetType attributeList;
};
```

*y sustitúyanse por las siguientes:*

```
valuetype CreateErrorInfoType : ApplicationErrorInfoType {
    public MONameSetType  relatedObjects;
    public AttributeSetType attributeList;
};
```

*Suprímense las siguientes líneas:*

```
valuetype DeleteErrorInfoType : ApplicationErrorInfoType {
    public MOSetType      relatedObjects;
    public AttributeSetType attributeList;
};
```

*y sustitúyanse por las siguientes:*

```
valuetype DeleteErrorInfoType : ApplicationErrorInfoType {
    public MONameSetType  relatedObjects;
    public AttributeSetType attributeList;
};
```

*Suprímense las siguientes líneas:*

```
/** Declaración hacia adelante. CORBA se sirve de referencias
de tipo "objeto" para identificar los objetos. Se utilizan en vez de
ejemplares de objetos de ASN.1. Para las interfaces de gestión de red, todos
los objetos heredan de la interfaz "ManagedObject". */

interface ManagedObject;

/** MO Set is a set of ManagedObject references. */

typedef sequence <ManagedObject> MOSetType;

/** MO Seq is a sequence of ManagedObject references. */

typedef sequence <ManagedObject> MOSeqType;
```

*y sustitúyanse por las siguientes:*

```
/** Una secuencia de nombres se define simplemente como una secuencia de nombres.
*/

typedef sequence <NameType> NameSeqType;
```

*Después de las siguientes líneas:*

```
/** Un conjunto de nombres se define como una secuencia de nombres. */
```

```
typedef sequence <NameType> NameSetType;
```

*insértense las siguientes líneas:*

```
/** Un nombre de objeto gestionado es simplemente un nombre */
```

```
typedef NameType MOnameType;
```

```
/** Una secuencia de nombres de objetos gestionados se define simplemente como una secuencia de nombres. */
```

```
typedef NameSeqType MOnameSeqType;
```

```
/** Un conjunto de nombres de objetos gestionados se define simplemente como un conjunto de nombres. */
```

```
typedef NameSetType MOnameSetType;
```

#### 14) *Nuevo Anexo C*

*Añádase el siguiente nuevo anexo para definir tipos comunes.*

### ANEXO C

#### El módulo IDL CORBA para tipos comunes

```
#ifndef _itut_x780ct_idl_  
#define _itut_x780ct_idl_
```

```
#include <itut_x780.idl>  
#include <itut_x780_1.idl>
```

```
#pragma prefix "itu.int"
```

```
/**
```

```
Este código IDL se ha de almacenar en un fichero llamado "itut_x780ct.idl" situado en el trayecto de búsqueda utilizado por compiladores IDL en el sistema del usuario.
```

```
*/
```

```
/**
```

```
Este módulo, itut_x780ct, contiene un tipo de datos común para el modelo de información basado en el marco CORBA del UIT-T. Este módulo se puede ampliar añadiendo nuevos tipos de datos.
```

```
*/
```

```
module itut_x780ct
```

```
{
```

#### **// Imports**

```
/**
```

```
Tipos importados de itut_x780
```

```
*/
```

```
typedef itut_x780::GeneralizedTimeType GeneralizedTimeType;
```

```
typedef itut_x780::PerceivedSeverityType PerceivedSeverityType;
```

#### **// Tipos de datos (X.721)**

```
struct CounterThresholdType
```

```
{
```

```
    long                comparisonLevel;
```

```

        long            offsetValue;
        boolean        notificationOnOff;
};

struct NotifyThresholdType
{
    float            threshold;
    boolean        notifyOnOff;
};

struct GaugeThresholdType
{
    NotifyThresholdType    notifyLow;
    NotifyThresholdType    notifyHigh;
};

typedef sequence<GaugeThresholdType> GaugeThresholdSetType;

enum TideMarkChoice
{
    tideMarkChoiceMax,
    tideMarkChoiceMin
};

union TideMarkType switch (TideMarkChoice)
{
    case tideMarkChoiceMax:
        float            maxTideMark;
    case tideMarkChoiceMin:
        float            minTideMark;
};

struct TideMarkInfoType
{
    TideMarkType            currentTideMark;
    TideMarkType            previousTideMark;
    GeneralizedTimeType    resetTime;
};

```

## // Tipos de datos (X.739)

/\*\*  
 PerceivedSeverityTypeOpt es un tipo facultativo. Si el discriminador es true (verdadero),  
 el valor está presente; de lo contrario el valor es nulo.  
 \*/

```

union PerceivedSeverityTypeOpt switch (boolean)
{
    case TRUE:
        PerceivedSeverityType    val;
};

```

/\*\*  
 SeverityIndicatingThresholdType contiene el nivel de umbral, que se debe aplicar  
 al atributo counter/gauge (contador/calibre). Debe ser inicializado cuando se crea el  
 objeto gestionado en el que está incluido, y se puede modificar. Se utiliza  
 un parámetro facultativo para asociar el nivel de umbral al parámetro severity (severidad)  
 de la notificación emitida. El parámetro booleano notifyOnOff permite desactivar la  
 generación de una notificación. El parámetro severity es obligatorio si notifyOnOff es  
 verdadero.

Si un conmutador notify-high o notify-low está activado (true) y el valor de indicación de  
 severidad no está presente, el valor indeterminado se envía con la  
 notificación, a menos que el comportamiento del objeto gestionado especifique otro valor.  
 \*/

```

struct SeverityIndicatingThresholdType
{
    float            threshold;
    boolean        notifyOnOff;
    PerceivedSeverityTypeOpt    severityIndication;
};

```

/\*\*

El atributo de SeverityIndicatingGaugeThresholdType tiene un comportamiento similar al del atributo gauge-threshold definido en X.721. La sintaxis tiene un parámetro adicional que indica el nivel de severidad asociado, definido en SeverityIndicatingThresholdType, para la notificación que es activada al rebasarse el correspondiente nivel de umbral. Completa la sintaxis del tipo de atributo gauge-threshold añadiendo un parámetro facultativo de indicación de severidad a la sintaxis de los dos submiembros notify-high y notify-low de cada miembro del nivel de umbral. Este tipo de atributo tiene un comportamiento adicional asociado a estos parámetros facultativos de indicación de severidad percibida, que se define de esta forma:

- Si el conmutador notify-high está activado (true), el valor de indicación de severidad de notify-high será comunicado en el parámetro de severidad percibida de una notificación generada cuando el valor de calibre cruza el valor umbral de calibre de notify-high en sentido positivo.

- Si el conmutador notify-low está activado (true), el valor de indicación de severidad de notify-low será comunicado en el parámetro de severidad percibida de una notificación generada cuando el valor de calibre cruza el umbral de calibre de notify-low en sentido negativo.

- Si los dos conmutadores están activados (true) para un solo valor de umbral, uno de los valores de indicación de severidad será "clear". El calibre-umbral que indica la severidad sólo emitirá una notificación de evento "clear" si la correspondiente notificación de nivel de umbral (sea notify-high o notify-low) ha sido emitida, y no se ha emitido ninguna otra notificación de "clear" para este par de niveles de umbral desde que se emitió la anterior notificación de nivel de umbral correspondiente.

\*/

```
struct SeverityIndicatingGaugeThresholdType
{
    SeverityIndicatingThresholdType notifyLow;
    SeverityIndicatingThresholdType notifyHigh;
};
```

/\*\*

El orden no es significativo.

\*/

```
typedef sequence<SeverityIndicatingGaugeThresholdType>
    SeverityIndicatingGaugeThresholdSetType;
```

```
struct TimeIntervalType
{
    unsigned short day;
    unsigned short hour;
    unsigned short minute;
    unsigned short second;
    unsigned short ms;
};
```

/\*\*

Las siguientes definiciones se han traducido a partir de definiciones ASN.1 en X.739.

\*/

```
enum TimePeriodChoice
{
    timePeriodChoiceDays,
    timePeriodChoiceHours,
    timePeriodChoiceMinutes,
    timePeriodChoiceSeconds,
    timePeriodChoiceMilliSeconds,
    timePeriodChoiceMicroSeconds,
    timePeriodChoiceNanoSeconds,
    timePeriodChoicePicoSeconds
};

union TimePeriodType switch (TimePeriodChoice)
{
    case timePeriodChoiceDays:
        long day;
    case timePeriodChoiceHours:
        long hour;
```

```
        case timePeriodChoiceMinutes:
            long minute;
        case timePeriodChoiceSeconds:
            long second;
        case timePeriodChoiceMilliSeconds:
            long ms;
        case timePeriodChoiceMicroSeconds:
            long us;
        case timePeriodChoiceNanoSeconds:
            long ns;
        case timePeriodChoicePicoSeconds:
            long ps;
    };
}; // module itut_x780ct
#endif // _itut_x780ct_idl_
```



## 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 multimedia
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedia
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
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
<b>Serie X</b>	<b>Redes de datos y comunicación entre sistemas abiertos</b>
Serie Y	Infraestructura mundial de la información y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación