

Unión Internacional de Telecomunicaciones

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**Z.141**

(02/2003)

SERIE Z: LENGUAJES Y ASPECTOS GENERALES DE  
SOPORTE LÓGICO PARA SISTEMAS DE  
TELECOMUNICACIÓN

Técnicas de descripción formal – Notación de prueba y  
de control de prueba

---

**Notación de pruebas y de control de pruebas  
versión 3: Formato de presentación tabular**

Recomendación UIT-T Z.141

UIT-T



RECOMENDACIONES UIT-T DE LA SERIE Z  
**Lenguajes y Aspectos Generales de Soporte Lógico para Sistemas de  
Telecomunicación**

<b>TÉCNICAS DE DESCRIPCIÓN FORMAL</b>	
Lenguaje de especificación y descripción	Z.100–Z.109
Aplicación de técnicas de descripción formal	Z.110–Z.119
Gráficos de secuencias de mensajes	Z.120–Z.129
Lenguaje ampliado de definición de objetos	Z.130–Z.139
<b>Notación de prueba y de control de prueba</b>	<b>Z.140–Z.149</b>
Notación de requisitos de usuarios	Z.150–Z.159
<b>Lenguajes de Programación</b>	
CHILL: el lenguaje de alto nivel del UIT-T	Z.200–Z.209
<b>Lenguaje Hombre-Máquina</b>	
Principios generales	Z.300–Z.309
Sintaxis básica y procedimientos de diálogo	Z.310–Z.319
LHM ampliado para terminales con pantalla de visualización	Z.320–Z.329
Especificación de la interfaz hombre-máquina	Z.330–Z.349
Interfaces hombre-máquina orientadas a datos	Z.350–Z.359
Interfaces hombre-máquina para la gestión de las redes de telecomunicaciones	Z.360–Z.379
<b>CALIDAD</b>	
Calidad de soportes lógicos de telecomunicaciones	Z.400–Z.409
Aspectos de la calidad de las Recomendaciones relativas a los protocolos	Z.450–Z.459
<b>MÉTODOS</b>	
Métodos para validación y pruebas	Z.500–Z.519
<b>SOPORTE INTERMEDIO</b>	
Entorno del procesamiento distribuido	Z.600–Z.609

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

## Recomendación UIT-T Z.141

### Notación de pruebas y de control de pruebas versión 3: Formato de presentación tabular

#### Resumen

En esta Recomendación se define el TFT, el formato tabular para la notación de pruebas y de control de pruebas versión 3 (TTCN-3). El TFT es el formato de presentación tabular del lenguaje núcleo de notación de pruebas y de control de pruebas versión 3 (TTCN-3, *testing and test control notation 3*) que se define en la Rec. UIT-T Z.140. Es similar en apariencia y funcionalidad al TTCN-2 definido en la Rec. UIT-T X.292 para la realización de las pruebas de conformidad. El formato tabular es una manera alternativa de visualizar el lenguaje núcleo que, al mismo tiempo, enfatiza los aspectos específicos de las sucesiones de pruebas de conformidad normalizadas.

Si bien el lenguaje núcleo puede utilizarse independientemente del formato de presentación tabular, este último no puede utilizarse sin el lenguaje núcleo. La utilización y aplicación del formato de presentación tabular se hará sobre la base del lenguaje núcleo.

Esta Recomendación define las proformas, las correspondencias de sintaxis, la semántica estática adicional, las restricciones de la semántica operacional, la visualización y otros atributos que, en conjunto, constituyen el formato de presentación tabular.

El TFT ha heredado todas las propiedades esenciales del lenguaje núcleo y está previsto para ser utilizado en las especificaciones de sucesiones de pruebas independientes de las plataformas, métodos de pruebas, capas de protocolos y protocolos. El TTCN-3 puede utilizarse para especificar todo tipo de pruebas de sistemas reactivas en diversos puertos de comunicaciones. Las esferas de aplicación típicas son las pruebas de protocolos (incluidos los protocolos móviles e Internet), las pruebas de servicio (incluidos los servicios suplementarios), las pruebas de módulo, las pruebas de plataformas CORBA y API. Queda fuera del alcance de esta Recomendación la especificación de sucesiones de pruebas para los protocolos de capa física.

Desde la primera publicación de las Recs. UIT-T Z.140 y Z.141 en julio de 2001, se han aportado actualizaciones significativas al lenguaje núcleo de TTCN-3, que se reflejan en el TFT. En esta Recomendación se incluyen los siguientes cambios:

- 1) correcciones a los ejemplos y otras correcciones formales;
- 2) reestructuración del documento para una mejor comprensión;
- 3) el anexo B, Semántica operativa, es ahora un documento independiente;
- 4) se han ampliado las funciones definidas por el usuario;
- 5) se han corregido los errores de la BNF, a la que también se han aportado las modificaciones resultantes de las actualizaciones;
- 6) se han añadido mecanismos de correspondencia de modelos;
- 7) se ha mejorado el sistema de tipos (incluida una mejor definición de la equivalencia de tipos) y se ha añadido un nuevo tipo de carácter;
- 8) se ha mejorado el mecanismo de importación; y
- 9) los alts nombrados se han eliminado y han sido sustituidos por altsteps, con una semántica mejorada.

#### Orígenes

La Recomendación UIT-T Z.141 fue aprobada el 13 de febrero de 2003 por la Comisión de Estudio 17 (2001-2004) del UIT-T por el procedimiento de la Recomendación UIT-T A.8.

La presente Recomendación es parte de una serie de Recomendaciones que tratan la notación de pruebas y de control de pruebas versión 3, como sigue:

Z.140: "TTCN-3: lenguaje núcleo";

**Z.141: "TTCN-3: formato de presentación tabular";**

Z.142: "TTCN-3: formato de presentación gráfico".

## 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

	<b>Página</b>
1 Alcance .....	1
2 Referencias .....	1
3 Siglas .....	1
4 Introducción.....	1
5 Convenios .....	2
5.1 Metanotación sintáctica .....	2
5.2 Texto de la especificación .....	3
5.3 Proformas .....	3
5.4 Lenguaje núcleo.....	3
5.5 Reglas generales de correspondencia .....	3
6 Proformas.....	4
6.1 Control de sucesión de pruebas .....	4
6.2 Parámetros de sucesión de pruebas .....	6
6.3 Importación de módulos .....	7
6.4 Tipos simples.....	8
6.5 Tipos estructurados.....	9
6.6 Tipos SequenceOf (secuencia de) .....	11
6.7 Tipo enumerado.....	12
6.8 Tipos de puertos .....	13
6.9 Tipos de componentes .....	14
6.10 Constantes.....	16
6.11 Firma.....	17
6.12 Plantillas simples .....	18
6.13 Plantilla estructurada .....	19
6.14 Función.....	21
6.15 Altstep.....	23
6.16 Caso de prueba .....	25
7 Producciones BNF .....	28



## Recomendación UIT-T Z.141

### Notación de pruebas y de control de pruebas versión 3: Formato de presentación tabular

#### 1 Alcance

Esta Recomendación define el formato de presentación tabular del TTCN versión 3 (o TTCN-3). Esta Recomendación se basa en el lenguaje núcleo TTCN-3 definido en la Rec. UIT-T Z.140.

La especificación de otros formatos queda fuera del alcance de esta Recomendación.

#### 2 Referencias

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. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

- [1] Recomendación UIT-T Z.140 (2003), *Notación de pruebas y de control de pruebas versión 3: Lenguaje núcleo*. Esta Recomendación también puede consultarse como norma del ETSI denominada SE 201 873-1 V2.2.1 (2002-09)
- [2] ETSI ES 201 873-4 (2003), *Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 4: TTCN-3 Operational Semantics*.

#### 3 Siglas

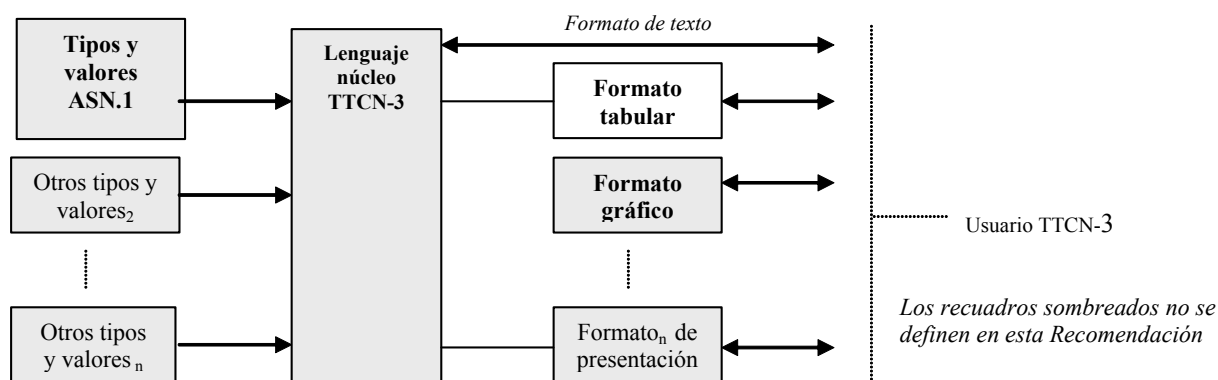
En la presente Recomendación se utilizan las siguientes siglas.

- ASN.1 Notación de sintaxis abstracta uno (*abstract syntax notation one*)
- ATS Sucesión de pruebas abstractas (*abstract test suite*)
- BNF Forma Backus-Naur (*Backus-Naur Form*)
- MTC Componente de prueba principal (*master test component*)
- PICS Declaración de conformidad de implementación de protocolo (*protocol implementation conformance statement*)
- PIXIT Información suplementaria de implementación de protocolo para pruebas (*protocol implementation extra information for testing*)
- TTCN Notación de pruebas y de control de pruebas (*testing and test control notation*)

#### 4 Introducción

El formato de presentación tabular para TTCN-3 (TFT) es un formato gráfico similar en apariencia y funcionalidad a versiones precedentes de TTCN, orientadas a la realización de pruebas de conformidad. El lenguaje núcleo de TTCN-3 se define en la Rec. UIT-T Z.140; proporciona una sintaxis completa textual, semántica estática y define asimismo la utilización del lenguaje con ASN.1. La semántica operacional es la definida en ES 201 873-4 [2]. El formato tabular

proporciona una forma alternativa de visualizar el lenguaje núcleo así como de enfatizar los aspectos específicos de los requisitos de una sucesión de pruebas de conformidad normalizada.



**Figura 1/Z.141 – Visión de usuario del lenguaje núcleo y de los diversos formatos de presentación**

El lenguaje núcleo puede utilizarse con independencia del formato de presentación tabular. Sin embargo, el formato tabular no puede utilizarse sin el lenguaje núcleo. La utilización e implementación del formato de presentación tabular se realizará sobre la base del lenguaje núcleo.

Esta Recomendación define:

- a) las proformas;
- b) las correspondencias de sintaxis;
- c) la semántica estática adicional;
- d) las restricciones de la semántica operacional;
- e) la visualización y otros atributos.

El conjunto de todas estas características constituye el formato de presentación tabular.

## 5 Convenios

En esta cláusula se definen los convenios utilizados en la definición de las proformas TTCN y en la gramática de lenguaje núcleo TTCN.

### 5.1 Metanotación sintáctica

En el cuadro 1 se define la metanotación utilizada para especificar la gramática BNF ampliada para TTCN (en adelante denominada BNF).



## Cuadro 1/Z.141 – Metanotación sintáctica TTCN.MP

::=	se define como
abc xyz	abc seguido de xyz
	alternativa
[abc]	0 ó 1 instancias de abc
{abc}	0 o más instancias de abc
{abc}+	1 o más instancias de abc
( ... )	agrupación textual
abc	el símbolo no terminal abc
<b>abc</b>	un símbolo terminal abc
"abc"	un símbolo terminal abc

En la cláusula 7 se definen las producciones de BNF. Las producciones no definidas en la cláusula 7 pueden encontrarse en el anexo A/Z.140.

### 5.2 Texto de la especificación

- En los campos de las proformas, para las referencias se utilizará **texto en negrita**.
- En las producciones BNF de lenguaje núcleo TTCN-3, para las referencias se utilizará *texto en cursiva*.
- Las palabras clave del lenguaje núcleo se indicarán en **texto en negrita courier new**.

### 5.3 Proformas

- En cada cuadro de un módulo TTCN-3 aparecerá literalmente **texto en negrita**.
- En los módulos TTCN-3 no aparecerá literalmente *texto en cursiva*. Este tipo de letra se utiliza para indicar que el texto existente será sustituido por el símbolo en letra cursiva. Los requisitos de sintaxis del texto existente pueden encontrarse a continuación de la definición de la proforma o en la BNF del lenguaje núcleo TTCN-3. Los corchetes antes y después de un *texto en cursiva* indican que la inclusión del texto en un determinado campo de la proforma es opcional.

### 5.4 Lenguaje núcleo

- Se reserva para las palabras clave y los terminales del lenguaje núcleo el **texto en negrita** de los caracteres entre comillas (por ejemplo, "{").
- En un módulo TTCN-3 no aparecerá literalmente *texto en cursiva*. Este tipo de letras se utiliza para indicar que el texto existente será sustituido por el símbolo en letra cursiva. Los requisitos de sintaxis del texto existente pueden encontrarse a continuación de la definición de la proforma o en la BNF del lenguaje núcleo TTCN-3.
- La notación "... " puede albergar cualquier contenido arbitrario que no se muestra explícitamente.

### 5.5 Reglas generales de correspondencia

La correspondencia entre el formato de presentación tabular y el lenguaje núcleo TTCN-3 se compone de un conjunto de transformaciones. Para todos los elementos sintácticos de cada proforma existe una transformación asociada. Las transformaciones también posibilitan la transformación de cualquier módulo de lenguaje núcleo en una representación tabular.

Estas transformaciones pueden ser de dos clases. La primera clase convierte directamente un elemento tabular en una construcción de lenguaje núcleo con el mismo significado. La segunda clase convierte un elemento tabular en una construcción de lenguaje núcleo asociada que no tiene significado a nivel de lenguaje núcleo.

Un ejemplo típico de la primera clase de transformación sería un campo identificador. Este campo puede ser transformado directamente de tabular a lenguaje núcleo manteniendo su significado, es decir, identificando algún elemento del lenguaje.

La segunda clase de transformación es típicamente alguna forma de comentario o directriz relativo a cómo debería representarse un elemento de lenguaje en el formato de presentación. Estos elementos no tienen un significado directo en el lenguaje núcleo y se expresan utilizando el enunciado *WithStatement*.

La sintaxis y la semántica que se especifican en esta Recomendación son propias del formato de presentación tabular. A fin de identificar inequívocamente en el lenguaje núcleo cuál es el formato de presentación que se está utilizando, se debería especificar el siguiente enunciado de visualización especial como primer enunciado de visualización asociado al módulo de lenguaje núcleo TTCN-3:

```

1: module TTCN3ModuleId "{"
2:   ...
3:   "}" with "{"
4:     display "" "presentation format" "==" "ETSI Tabular version"
5:     MajorVersion "." MinorVersion "" ";"
6:     ...
7:   "}"

```

NOTA – Todos los enunciados *WithStatements* asociados con una proforma determinada deberán agruparse en una lista contigua.

Los campos **Group** de las proformas nunca se traducen en *WithStatements* sino que se derivan de la estructura de grupo existente en la especificación de módulo.

## 6 Proformas

### 6.1 Control de sucesión de pruebas

Control de sucesión de pruebas			
Nombre de módulo	TTCN3ModuleId		
Versión	[TabeeText]		
Fecha	[TabFreeText]		
Ref. normalizada de base	[TabFreeText]		
Ref. normalizada de prueba	[TabFreeText]		
Ref. PICS	[TabFreeText]		
Ref. PIXIT	[TabFreeText]		
Método(s) de prueba	[TabFreeText]		
Codificación	[TabFreeText]		
Comentarios	[TabFreeText]		
Nombre de definición local	Tipo	Valor inicial	Comentarios
[VarConstOrTimerIdentifier]	[ConstTypeOrTimer]	[Expression]	[TabFreeText]
...	...	...	...
Comportamiento			
ModuleControlBody			
Comentarios detallados	[TabFreeText]		

Figura 2/Z.141 – Proforma del control de sucesión de pruebas

## 6.1.1 Correspondencia

La proforma del control de sucesión de pruebas se traduce en tres partes. La primera parte está formada por los campos de encabezamiento y el campo **comentarios detallados**, que se convierten en atributos de visualización en el enunciado *WithStatement* asociado con el módulo TTCN-3 general. Se establece una correspondencia entre el campo **nombre de módulo** y el identificador de módulo.

La segunda parte está formada por constantes, variables y temporizadores locales definidos en la parte de control. Estas definiciones pueden aparecer en cualquier lugar de la parte de control del lenguaje núcleo, pero en la proforma están separados del resto del cuerpo principal del módulo y se presentan en un cuadro distinto. Ha de respetarse el orden de las definiciones, ya que pueden ser interdependientes. En la columna **tipo** se pondrá la palabra clave **temporizador** para todos los temporizadores y, para las constantes, se indicará el tipo de constante precedido de la palabra clave **const**. El campo **comentarios** del cuadro de definiciones locales se convierte en atributos de visualización en el enunciado *WithStatement* asociado con la parte de control del módulo de lenguaje núcleo TTCN-3.

La tercera parte es el control del módulo de lenguaje núcleo TTCN-3, excepto las constantes, variables y temporizadores locales.

```

1:   module TTCN3ModuleId "{"
2:       control "{"
3:       var Type VarIdentifier [":=" Expression] ";"
4:       timer TimerIdentifier [":=" Expression] ";"
5:       const Type ConstIdentifier [":=" ConstantExpression];
6:       ModuleControlBody
7:   }" with "{"
8:       { VarConstOrTimerCommentsAttribute }
9:   }"
10:  }" with "{"
11:  ModuleAttributes
12:  [EncodeAttribute;]
13:  }"

```

Ejemplo:

Control de sucesión de pruebas			
<b>Nombre de módulo</b>	Example1		
<b>Versión</b>	1.01		
<b>Fecha</b>	19 de julio de 2001		
<b>Ref. de normas de base</b>	Recomendación UIT-T Q.123		
<b>Ref. de normas de prueba</b>	Recomendación UIT-T Q.123.1		
<b>Ref. PICS</b>	Recomendación UIT-T Q.123.2, anexo A		
<b>Ref. PIXIT</b>	Recomendación UIT-T Q.123.2, anexo B		
<b>Método(s) de prueba</b>	local		
<b>Codificación</b>	BER		
<b>Comentarios</b>	ATS escrito por STF 133		
Nombre de definición local	Tipo	Valor inicial	Comentarios
PI	const float	3.14	relación
X	float	PI * 2	doble PI
t1	timer	15	temporizador de 15 segundos
Comportamiento			
<pre> /* group1/ */ /* group1_1/ */   execute(test1);   execute(test2); /* group1_2/ */   execute(test3);   execute(test4); /* group2/ */   execute(test5); </pre>			
<b>Comentarios detallados</b>	comentarios detallados		

Se corresponde con:

```

1:  module Example1 {
2:      control {
3:          const float PI := 3.14;
4:          var float x := PI * 2;
5:          timer t1 := 15;
6:
7:          /* group1/ */
8:              /* group1_1/ */
9:                  execute(test1());
10:                 execute(test2());
11:             /* group1_2/ */
12:                 execute(test3());
13:                 execute(test4());
14:             /* group2/ */
15:                 execute(test5());
16:         } with {
17:             display (PI) "comments := the ratio";
18:             display (x) "comments := double PI";
19:             display (t1) "comments := a 15 second timer";
20:         }
21:     } with {
22:         display "presentation format := ETSI Tabular version 1.0";
23:         display "module version := 1.01";
24:         display "module date := 19 July 2001";
25:         display "module base standards ref := ITU-T Recommendation Q.123";
26:         display "module test standards ref := ITU-T Recommendation Q.123";
27:         display "module pics ref := ITU-T Recommendation Q.123 Annex A";
28:         display "module pixit ref := ITU-T Recommendation Q.123 Annex A";
29:         display "module test method := local";
30:         display "module comments := ATS written by STF 133";
31:         display "module detailed comments := detailed comments";
32:         encode "BER";
33:     }

```

## 6.2 Parámetros de sucesión de pruebas

Parámetros de sucesión de pruebas				
Nombre	Tipo	Valor inicial	Ref. PICS/PIXIT	Comentarios
.	.	.	.	.
<i>ModuleParIdentifier</i>	<i>ModuleParType</i>	<i>[ConstantExpression]</i>	<i>[TabFreeText]</i>	<i>[TabFreeText]</i>
.	.	.	.	.
<b>Comentarios detallados</b>	<i>[TabFreeText]</i>			

Figura 3/Z.141 – Proforma de los parámetros de sucesión de pruebas

### 6.2.1 Correspondencia

Se establece una correspondencia entre todas las entradas de la proforma de parámetros de sucesión de pruebas y las *ModuleParList* del *ModuleParameterDefs* del módulo TTCN-3 asociado. Si hay más de un *ModuleParameterDef*, se recopilan todas las *ModuleParList* y se representan en una única proforma de **parámetros de sucesión de pruebas**.

Se establece una correspondencia entre los campos **ref. PICS/PIXIT** y **comentarios** y los atributos de visualización definidos por el identificador de parámetro del enunciado *WithStatements* asociado con el *ParamDef* circundante. Se establece una correspondencia entre el campo **comentarios detallados** y un atributo de visualización del enunciado *WithStatement* asociado con el *ParamDef* circundante.

```

1: module TTCN3ModuleId "{"
2:   parameters "{" ModuleParList "}"
3:   with "{"
4:     [ModuleParPicsPixitRefAttribute ";"]
5:     [ModuleParComments ";"]
6:     [DetailedComments ";"]
7:   "}"
8: "}"

```

Ejemplo:

Parámetros de sucesión de pruebas				
Nombre	Tipo	Valor inicial	Ref. PICS/PIXIT	Comentarios
CAP_1	Boolean	true	A.1.3	opción 1 aplicada
Tall	Float	600.0	A.1.4	temporizador general del módulo
<b>Comentarios detallados</b>	comentarios detallados			

Se corresponde con:

```

1: module MyModule{
2:   parameters { boolean CAP_1 := true, float Tall := 600.0 }
3:   with {
4:     display (CAP_1) "pics/pixit ref := A.1.3";
5:     display (CAP_1) "comments := option 1 implemented";
6:     display (Tall) "pics/pixit ref := A.1.4";
7:     display (Tall) "comments := overall module timer";
8:     display "detailed comments := detailed comments";
9:   }
10: }

```

### 6.3 Importación de módulos

Importaciones	
Nombre de fuente	GlobalModuleId [ <i>recursive</i> ]
Lenguaje de fuente	[LanguageSpec]
Grupo	[GroupReference]
Ref. de fuente	[TabFreeText]
Codificación	[TabFreeText]
Comentarios	[TabFreeText]
<b>Tipo</b>	<b>Nombre</b>
[ImportType]	ImportSpecification
<b>Comentarios detallados</b>	[TabFreeText]

Figura 4/Z.141 – Proforma de importaciones

#### 6.3.1 Correspondencia

Se establece una correspondencia entre la proforma de importaciones y un enunciado *ImportDef* del lenguaje núcleo TTCN-3. Los campos **nombre de fuente**, **lenguaje de fuente**, **tipo** y **nombre** se utilizan directamente en el correspondiente enunciado *ImportDef* del lenguaje núcleo. Los campos **ref. de fuente**, **comentarios** y **comentarios detallados** se traducen en atributos de visualización en el enunciado *WithStatement* asociado con el enunciado *ImportDef*. El campo **codificación** se traduce en un atributo de codificación en el enunciado *WithStatement* asociado con el enunciado *ImportDef*.

Si se importan todas las definiciones de un módulo, *ImportType* estará vacío y en la *ImportSpecification* se utilizará la palabra clave **all**.

```

1: module TTCN3ModuleId "{"
2:   ImportDef
3:   with "{"
4:     [ImportsSourceRefAttribute ";"]
5:     [CommentsAttribute ";"]
6:     [ImportsSourceDefinitionCommentsAttribute ";"]
7:     [DetailedCommentsAttribute ";"]
8:     [EncodeAttribute ";"]
9:   }"
10: }"

```

Ejemplo:

Importaciones		
Nombre de fuente	ModuleA recursive	
Lenguaje de fuente	ASN.1:1997	
Grupo		
Ref. de fuente	EN 800 900 version 2	
Codificación	BER	
Comentarios	Declaraciones de importación desde un ATS	
Tipo	Nombre	Comentarios
Constante	todos, excepto foobar	
Tipo	MyType	foobar
Grupo	AtoU_CTR	
Comentarios detallados	Comentarios detallados	

Se corresponde con:

```

1: module MyModule {
2:   import from ModuleA recursive language "ASN.1:1997" {
3:     const all except foobar;
4:     type MyType;
5:     Group AtoU_CTR;
6:   } with {
7:     display "imports source ref := EN 800 900 version 2";
8:     display "comments := importing declarations from ATS";
9:     display "detailed comments := detailed comments";
10:    encode "BER";
11:  }
12: }

```

## 6.4 Tipos simples

Tipos simples			
Grupo	[GroupReference]		
Nombre	Definición	Codificación	Comentarios
<i>SubTypeIdentifier</i>	<i>Type</i> [ArrayDef] [SubTypeSpec]	[TabFreeText]	[TabFreeText]
Comentarios detallados	[TabFreeText]		

Figura 5/Z.141 – Proforma de tipos simples

### 6.4.1 Correspondencia

Se establece una correspondencia entre la proforma de tipos simples y un conjunto de enunciados de definición de tipos simples en el mismo nivel de grupo. Las definiciones de tipos simples son todas del tipo *SubTypeDef*.

Se establece una correspondencia entre el campo **comentarios detallados** y un atributo de visualización del enunciado *WithStatement* asociado con el grupo o el módulo circundante. Se establece una correspondencia entre los campos **codificación** y **comentarios** y los atributos de codificación y visualización, respectivamente, del enunciado *WithStatement* asociado con la definición de tipo simple correspondiente.

```

1: module TTCN3ModuleId "{"
2:   type Type SubTypeIdentifier [ArrayDef] [SubTypeSpec] with "{"
3:     [EncodeAttribute ";""]
4:     [CommentsAttribute ";""]
5:   }" with "{"
6:     [SimpleTypesDetailedCommentsAttribute ";""]
7:   }"
8: }"

```

Ejemplo:

Tipos simples			
Grupo	SimpleTypes/		
Nombre	Definición	Codificación	Comentarios
EQ_NUMBER	entero (1 .. 20)	PER	A saber
<b>Comentarios detallados</b>	comentarios detallados		

Se corresponde con:

```

1: module MyModule {
2:   group SimpleTypes {
3:     type integer EQ_NUMBER (1..20) with {
4:       encode "PER";
5:       display "comments := God knows";
6:     }
7:   } with {
8:     display "simple types detailed comments := detailed comments";
9:   }
10: }

```

## 6.5 Tipos estructurados

Tipo estructurado			
<b>Nombre</b>	<i>StructTypeIdentifier</i> [ <i>StructDefFormalParList</i> ]		
<b>Grupo</b>	[ <i>GroupReference</i> ]		
<b>Estructura</b>	<i>StructureType</i>		
<b>Codificación</b>	[ <i>TabFreeText</i> ]		
<b>Comentarios</b>	[ <i>TabFreeText</i> ]		
Nombre de campo	Tipo de campo	Codificación de campo	Comentarios
.	.	.	.
<i>FieldIdentifier</i>	<i>Type</i> [ <i>ArrayDef</i> ]	[ <i>TabFreeText</i> ]	[ <i>TabFreeText</i> ]
.	[ <i>SubTypeSpec</i> ]	.	.
.	[ <i>OptionalKeyword</i> ]	.	.
.	.	.	.
<b>Comentarios detallados</b>	[ <i>TabFreeText</i> ]		

Figura 6/Z.141 – Proforma de tipo estructurado

### 6.5.1 Correspondencia

Se establece una correspondencia entre la proforma de tipo estructurado y el enunciado de definición de tipo estructurado en TTCN-3. Los tipos *RecordDef*, *UnionDef* y *SetDef* utilizan esta proforma.

Se establece una correspondencia entre los campos **comentarios** y **comentarios detallados** y los atributos de visualización del enunciado *WithStatement* correspondiente. Se establece la correspondencia entre el campo **codificación** y un atributo de codificación del enunciado *WithStatement* correspondiente. Se establece una correspondencia entre los campos **comentarios** y **codificación de campo** de cada elemento de campo y un atributo de visualización y de codificación, respectivamente, determinados por el *FieldIdentifier* del enunciado *WithStatement* correspondiente.

```

1: module TTCN3ModuleId "{"
2:   type StructureType StructTypeIdentifier [StructDefFormalParList] "{"
3:     {Type FieldIdentifier [ArrayDef] [SubtypeSpec] [OptionalKeyword]}
4:   "}" with "{"
5:     [EncodeAttribute ";"]
6:     [CommentsAttribute ";"]
7:     {FieldCommentsAttribute ";"}
8:     {FieldEncodeAttribute ";"}
9:     [DetailedCommentsAttribute ";"]
10:  "}"
11: "}"

```

Ejemplo:

Tipo estructurado			
<b>Nombre</b>	StructTypeIdentifier[StructDefFormalParList]		
<b>Grupo</b>	[GroupReference]		
<b>Estructura</b>	StructureType		
<b>Codificación</b>	[TabFreeText]		
<b>Comentarios</b>	[TabFreeText]		
Nombre de elemento	Definición de tipo	Codificación de campo	Comentarios
DestPC	BIT_14		código de punto de destino
OrigPC	BIT_14		código de punto de origen
SLSel	SLSel_Type	PER	selección de enlace de señalización
<b>Comentarios detallados</b>	anula las definiciones anteriores		

Se corresponde con:

```

1: module MyModule {
2:   type record routing_label(SLSel_Type) {
3:     BIT_14 DestPC,
4:     BIT_14 OrigPC,
5:     SLSel_Type SLSel
6:   } with {
7:     encode "BER";
8:     display "comments := header for routing info";
9:     display (DestPC) "comments := destination point code";
10:    display (OrigPC) "comments := origination point code";
11:    display (SLSel) "comments := signalling link selection";
12:    encode (SLSel) "PER";
13:    display "detailed comments := overrides previous definitions";
14:  }
15: }

```



## 6.6 Tipos SequenceOf (secuencia de)

Tipos SequenceOf					
Grupo	[GroupReference]				
Nombre	Tipo	Clase	Longitud	Codificación	Comentarios
<i>StructTypeIdentifier</i>	. Type [SubTypeSpec] .	. RecordOrSet .	. [StringLength] .	. [TabFreeText] .	. [TabFreeText] .
<b>Comentarios detallados</b>	[TabFreeText]				

Figura 7/Z.141 – Proforma de tipos SequenceOf

### 6.6.1 Correspondencia

Se establece una correspondencia entre la proforma de tipos SequenceOf y una serie de enunciados de definición de tipo SequenceOf del mismo nivel de grupo. Esta proforma se utilizará para las definiciones de los tipos *RecordOfDef* y *SetOfDef*.

Se establece una correspondencia entre el campo **comentarios detallados** y un atributo de visualización del enunciado *WithStatement* asociado con el grupo o el módulo circundante. Se establece una correspondencia entre los campos **codificación** y **comentarios** y los atributos de codificación y visualización, respectivamente, del enunciado *WithStatement* asociado con la definición de tipo SequenceOf respectiva.

```

1: module TTCN3ModuleId "{"
2:   type record of [StringLength] Type StructTypeIdentifier [SubTypeSpec]
3:     with "{"
4:       [EncodeAttribute ";"]
5:       [CommentsAttribute ";"]
6:     }"
7:   type set of [StringLength] Type StructTypeIdentifier [SubTypeSpec]
8:     with "{"
9:       [EncodeAttribute ";"]
10:      [CommentsAttribute ";"]
11:     }"
12:  }" with "{"
13:    [SequenceOfTypesDetailedCommentsAttribute ";"]
14:  }"

```

Ejemplo:

Tipos SequenceOf					
Grupo	SequenceOfTypes/				
Nombre	Tipo	Clase	Longitud	Codificación	Comentarios
RecordOfIntegers	Entero (1..10)	record	10	BER	Diez enteros
SetOfBooleans	Booleano	set	3	PER	Tres booleanos
<b>Comentarios detallados</b>	Ejemplos de tipos sequenceOf				

Se corresponde con:

```

1: module MyModule {
2:   group SequenceOfTypes {
3:     type record of length(10) integer RecordOfIntegers(1..10) with {
4:       encode "BER";
5:       display "comments := ten integers";
6:     }
7:     type set of length(3) boolean SetOfBooleans with {
8:       encode "PER";
9:       display "comments := three booleans";
10:    }
11:   } with {
12:     display "sequenceof types detailed comments := example sequenceof
13: types";
14:   }
15: }

```

## 6.7 Tipo enumerado

Tipo enumerado		
Nombre	<i>EnumTypeIdentifier</i>	
Grupo	[GroupReference]	
Codificación	[TabFreeText]	
Comentarios	[TabFreeText]	
Nombre de enumeración	Valor de enumeración	Comentarios
<i>EnumerationIdentifier</i>	[Number]	[TabFreeText]
Comentarios detallados	[TabFreeText]	

Figura 8/Z.141 – Proforma de tipo enumerado

### 6.7.1 Correspondencia

Se establece una correspondencia entre la proforma del tipo enumerado y un enunciado de definición de tipo enumerado en el lenguaje núcleo TTCN-3. Se establece una correspondencia entre los campos **comentarios** y **comentarios detallados** y los atributos de visualización del enunciado *WithStatement* correspondiente. Se establece una correspondencia entre el campo **codificación** y un atributo de codificación en el enunciado *WithStatement* correspondiente. Se establece una correspondencia entre los campos **comentarios** de cada enumeración y los atributos de visualización determinados por *EnumerationIdentifier* del enunciado *WithStatement* correspondiente.

```

1: module TTCN3ModuleId "{"
2:   type enumerated EnumTypeIdentifier "{"
3:     EnumerationIdentifier ["(" Number ")"]
4:     {" EnumerationIdentifier ["(" Number ")"]}
5:   }" with "{"
6:     [EncodeAttribute ";"]
7:     [CommentsAttribute ";"]
8:     {NamedValueCommentsAttribute ";"}
9:     [DetailedCommentsAttribute ";"]
10:  }"
11: }"

```

Ejemplo:

Tipo enumerado		
Nombre	Weekdays	
Grupo		
Codificación	BER	
Comentarios	días de la semana	
Nombre de enumeración	Valor de enumeración	Comentarios
Monday	1	
Tuesday	2	
Wednesday	3	Punto medio
Thursday	4	
Friday	5	TGIF
Saturday	6	
Sunday	7	
Comentarios detallados	Ojalá fuera viernes	

Se corresponde con:

```

1: module MyModule {
2:   type enumerated Weekdays {
3:     Monday(1), Tuesday(2), Wednesday(3), Thursday(4), Friday(5),
4:     Saturday(6), Sunday(7)
5:   } with {
6:     encode "BER";
7:     display "comments := days of the week";
8:     display (Wednesday) "comments := half way there";
9:     display (Friday) "comments := TGIF";
10:    display "detailed comments := wish it were Friday";
11:  }
12: }

```

## 6.8 Tipos de puertos

Tipo de puerto		
Nombre	<i>PortTypeIdentifier</i>	
Grupo	<i>[GroupReference]</i>	
Codificación	<i>PortModelType</i>	
Comentarios	<i>[TabFreeText]</i>	
Tipo/Firma	Dirección	Comentarios
<i>.</i> <i>TypeOrSignature</i> <i>.</i>	<i>.</i> <i>InOutOrInout</i> <i>.</i>	<i>.</i> <i>[TabFreeText]</i> <i>.</i>
Comentarios detallados	<i>[TabFreeText]</i>	

Figura 9/Z.141 – Proforma de tipo de puerto

### 6.8.1 Correspondencia

Se establece una correspondencia entre la proforma de tipo de puerto y una definición de tipo de puerto en el lenguaje núcleo TTCN-3. Se establece una correspondencia entre los campos **comentarios** y **comentarios detallados** y los atributos de visualización en el enunciado *WithStatement* correspondiente. Se establece una correspondencia entre los campos **comentarios** del cuadro tipos y firma y los atributos de visualización en el enunciado *WithStatement* correspondiente determinados por un identificador de tipo o firma. Habrá siempre una fila para cada tipo o firma.

En el campo **tipo/firma** se pone la palabra clave **all** si todos los tipos o firmas de procedimiento definidos en el módulo pueden pasar por el puerto de comunicación.

```

1: module TTCN3ModuleId "{"
2:   type port PortTypeIdentifier PortModelType "{"
3:     PortTypeDef
4:   }" with "{"
5:     [CommentsAttribute ";"]
6:     {TypeOrSignatureCommentsAttribute ";"}
7:     [DetailedCommentsAttribute ";"]
8:   }"
9: }"

```

Ejemplo:

Tipo de puerto		
<b>Nombre</b>	MyPortType	
<b>Grupo</b>		
<b>Codificación</b>	mensaje	
<b>Comentarios</b>	ejemplo de tipo de puerto	
Tipo/firma	Dirección	Comentarios
MsgType1	entrada	primer comentario
MsgType2	entrada	segundo comentario
MsgType3	salida	
<b>Comentarios detallados</b>	comentario detallado	

Se corresponde con:

```

1: module MyModule {
2:   type port MyPortType message {
3:     in MsgType1;
4:     in MsgType2;
5:     out MsgType3;
6:   } with {
7:     display "comments := example port type";
8:     display (MsgType1) "comments := first comment";
9:     display (MsgType2) "comments := second comment";
10:    display "detailed comments := detailed comment";
11:   }
12: }

```

## 6.9 Tipos de componentes

Tipo de componente			
<b>Nombre</b>	ComponentTypeIdentifier		
<b>Grupo</b>	[GroupReference]		
<b>Comentarios</b>	[TabFreeText]		
Nombre de definición local	Tipo	Valor inicial	Comentarios
.	.	.	.
VarConstOrTimerIdentifier	TypeOrTimer [ArrayDef]	[ConstantExpression   Expression]	[TabFreeText]
.	.	.	.
Nombre de puerto	Tipo de puerto		Comentarios
.	.		.
PortIdentifier	PortType [ArrayDef]		[TabFreeText]
.	.		.
<b>Comentarios detallados</b>	[TabFreeText]		

Figura 10/Z.141 – Proforma de tipo de componente

### 6.9.1 Correspondencia

Se establece una correspondencia entre la proforma de tipo de componente y una definición de tipo de componente en lenguaje núcleo TTCN-3. La proforma se traduce en tres partes.

La primera parte está formada por los campos de encabezamiento **comentarios** y **comentarios detallados**, que se convierten en atributos de visualización en el enunciado *WithStatement* asociado con la definición de tipo de componente.

La segunda parte está formada por constantes, variables y temporizadores locales definidos en el tipo de componente. Estas definiciones pueden aparecer en cualquier lugar de la definición tipo de componente en lenguaje núcleo, pero en la proforma se separan de las instancias de puerto y se visualizan en un cuadro distinto. Ha de respetarse el orden de las definiciones, ya que pueden ser interdependientes. En la columna **Tipo** se pondrá la palabra clave **timer** para todos los temporizadores, y el tipo de constante precedido de la palabra clave **const** para todas las constantes. Siempre habrá una fila para cada constante, variable o temporizador. La columna **comentarios** de este cuadro se convierte en atributos de visualización determinados por el identificador de definición local del enunciado *WithStatement* asociado con la definición de tipo de componente.

La tercera parte está formada por instancias de puerto definidas en el tipo de componente. Cualquier grupo de definiciones se anexa al tipo de puerto. Siempre habrá una fila para cada instancia de puerto. La columna **comentarios** de este cuadro se transforma en los atributos de visualización determinados por *PortIdentifier* en el enunciado *WithStatement* asociado con la definición de tipo de componente.

```

1: module TTCN3ModuleId "{"
2:   type component ComponentTypeIdentifier "{"
3:     var Type VarIdentifier [":=" Expression] ";"
4:     timer TimerIdentifier [":=" Expression] ";"
5:     const Type ConstIdentifier ":=" ConstantExpression ";"
6:     PortList
7:   }" with "{"
8:     [CommentsAttribute ";" ]
9:     {PortCommentsAttribute ";" }
10:    [DetailedCommentsAttribute ";" ]
11:   }"
12: }"

```

Ejemplo:

Tipo de componente			
<b>Nombre</b>	MyComponentType		
<b>Grupo</b>			
<b>Comentarios</b>	ejemplo de tipo de componente		
<b>Nombre de definición local</b>	<b>Tipo</b>	<b>Valor inicial</b>	<b>Comentarios</b>
PI	const float	3.14	relación
X	Float	PI * 2	doble PI
t1	Timer	15 min	temporizador de 15 segundos
<b>Nombre de puerto</b>		<b>Tipo de puerto</b>	<b>Comentarios</b>
PC01		MyMessagePortType	primer comentario
PC02		MyProcedurePortType	segundo comentario
<b>Comentarios detallados</b>	comentarios detallados		

Se corresponde con:

```

1: module MyModule {
2:   type component MyComponentType {
3:     const float PI := 3.14;
4:     var float x := PI * 2;
5:     timer t1 := 15;
6:     port MyMessagePortType PCO1;
7:     port MyProcedurePortType PCO2;
8:   } with {
9:     display "comments := an example component type";
10:    display (PI) "comments := the ratio";
11:    display (x) "comments := double PI";
12:    display (t1) "comments := a 15 second timer";
13:    display (PCO1) "comments := first comment";
14:    display (PCO2) "comments := second comment";
15:    display "detailed comments := detailed comments";
16:  }
17: }

```

## 6.10 Constantes

Constantes			
Grupo	[GroupReference]		
Nombre	Tipo	Valor	Comentarios
<i>ConstIdentifier</i> / <i>ExtConstIdentifier</i>	<i>Type</i> [ArrayDef]	<i>ConstantExpression</i> / <b>external</b>	[TabFreeText]
<b>Comentarios detallados</b>	[TabFreeText]		

Figura 11/Z.141 – Proforma de constantes

### 6.10.1 Correspondencia

Se establece una correspondencia entre la proforma de constantes y un grupo de enunciados de definiciones de constantes y constantes externas en el mismo nivel de grupo. Se establece una correspondencia entre el campo **comentarios detallados** y un atributo de visualización en el enunciado *WithStatement* asociado con el grupo o el módulo circundante. Se establece una correspondencia entre los campos **comentarios** y los atributos de visualización en el enunciado *WithStatement* asociado con las respectivas definiciones de constantes. En el caso de una constante externa, en el campo **valor** se pone la palabra clave **external**.

```

1: module TTCN3ModuleId "{"
2:   const Type ConstIdentifier[ArrayDef] " := " ConstantExpression with "{"
3:     [CommentsAttribute ";"]
4:   }"
5:   external const Type ConstIdentifier with "{"
6:     [CommentsAttribute ";"]
7:   }"
8: }" with "{"
9:   [ConstantsDetailedCommentsAttribute ";"]
10: }"

```

Ejemplo:

Constante			
Grupo	Constants1		
Nombre	Tipo	Valor	Comentarios
TOTO	entero	external	definido en otro lugar
SEL2	booleano	(5 + TOTO) < 10	se alcanza el límite TOTO
T1	entero[1..3]	{1,3,2}	
<b>Comentarios detallados</b>	comentarios detallados		

Se corresponde con:

```

1: module MyModule {
2:   group Constants1 {
3:     external const integer TOTO with {
4:       display "comments := defined somewhere else";
5:     }
6:     const boolean SEL2 := (5 + TOTO) < 10 with {
7:       display "comments := TOTO limit reached";
8:     }
9:     const integer T1[1..3] := {1,3,2};
10:  } with {
11:   display "detailed comments := detailed comments";
12:  }
13: }

```

## 6.11 Firma

Definición de firma	
<b>Nombre</b>	<i>SignatureIdentifier</i> ([ <i>SignatureFormalParList</i> ])
<b>Grupo</b>	[ <i>GroupReference</i> ]
<b>Tipo de retorno</b>	[ <i>Type</i> ]   <b>noblock</b>
<b>Comentarios</b>	[ <i>TabFreeText</i> ]
<b>Tipo de excepción</b>	
.	.
[ <i>ExceptionType</i> ]	[ <i>TabFreeText</i> ]
.	.
<b>Comentarios detallados</b>	[ <i>TabFreeText</i> ]

Figura 12/Z.141 – Proforma de definición de firma

### 6.11.1 Correspondencia

Se establece una correspondencia entre la proforma de definición de firma y una definición de firma en lenguaje núcleo TTCN-3. Se establece una correspondencia entre los campos **comentarios** y **comentarios detallados** y los atributos de visualización en el enunciado *WithStatement* correspondiente. Se establece una correspondencia entre el campo **comentarios** del cuadro de excepciones y los atributos de visualización definidos por el tipo de excepción en el enunciado *WithStatement* correspondiente. Los procedimientos de no bloqueo se especificarán mediante la palabra clave **noblock** como tipo de retorno.

```

1: module TTCN3ModuleId "{
2:   signature SignatureIdentifier "(" [SignatureFormalParList] ")"
3:   [return Type | noblock]
4:   [exception "(" ExceptionTypeList ")"]
5:   with "{"
6:     [CommentsAttribute ";"]
7:     [ExceptionCommentsAttribute ";"]
8:     [DetailedCommentsAttribute ";"]
9:   }"
10: }"

```

Ejemplo:

Definición de firma	
Nombre	read_syscall(integer fields, inout charstring buf, integer nbyte)
Grupo	
Tipo de retorno	entero
Comentarios	lectura desde un fichero
Tipo de excepción	Comentarios
Integer	código de error
MyException	definido por el usuario
Comentarios detallados	requerido: unistd.h

Se corresponde con:

```

1: module MyModule {
2:   signature read_syscall(in integer fields,
3:                         inout charstring buf,
4:                         in integer nbyte)
5:   return integer
6:   exception (integer, MyException)
7:   with {
8:     display "comments := reads from a file";
9:     display (integer) "comments := error code of system call";
10:    display (MyException) "comments := user defined";
11:    display "detailed comments := required: unistd.h";
12:  }
13: }

```

## 6.12 Plantillas simples

Plantillas simples					
Grupo	[GroupReference]				
Nombre	Tipo	Derivado	Valor	Codificación	Comentarios
.	.	.	.	.	.
TemplateIdentifier	BaseTemplate	[DerivedDef]	TemplateBody	[TabFreeText]	[TabFreeText]
.	.	.	.	.	.
Comentarios detallados	[TabFreeText]				

Figura 13/Z.141 – Proforma de plantillas simples

### 6.12.1 Correspondencia

Se establece una correspondencia entre la proforma de plantillas simples y un grupo de enunciados de definición de plantillas simples en el mismo nivel de grupo. Las definiciones de plantillas simples son siempre definiciones de plantillas que tienen *SimpleSpec* o *ArrayValueOrAttrib* como *TemplateBody*. Los tipos correspondientes se definen en un proforma de tipo simples, tipos *SequenceOf* o tipos enumerados.

Se establece una correspondencia entre el campo **comentarios detallados** y un atributo de visualización en el enunciado *WithStatement* asociado con el grupo a módulo circundante. Se establece una correspondencia entre los campos **comentarios** y **codificación** y los atributos de visualización y codificación determinados por *TemplateIdentifier* en el enunciado *WithStatement* asociado con el correspondiente enunciado de definición de plantilla simple.



```

1: module TTCN3ModuleId "{"
2:   template BaseTemplate[DerivedDef] := TemplateBody with "{"
3:     [EncodeAttribute ";"]
4:     [CommentsAttribute ";"]
5:   }"
6: }" with "{"
7:   [SimpleTemplatesDetailedCommentsAttribute ";"]
8: }"

```

Ejemplo:

Plantilla simple					
Grupo		SimpleTemplates1			
Nombre	Tipo	Derivado	Valor	Codificación	Comentarios
MyTemplate1	MyType1		3	BER	foobar
MyTemplate11 (integer index)	MyType1	MyTemplate1	3*index	PER	índice actual
Comentarios detallados		ejemplo			

Se corresponde con:

```

1: module MyModule {
2:   group SimpleTemplates {
3:     template MyType1 MyTemplate1 with {
4:       encode "BER";
5:       display "comments := foobar";
6:     }
7:     template MyType1 MyTemplate11(integer index)
8:       modifies MyTemplate1 := 3 * index
9:     with {
10:      encode "PER";
11:      display "comments := the current index";
12:    }
13:   } with {
14:     display "simple templates detailed comments := an example";
15:   }
16: }

```

### 6.13 Plantilla estructurada

Plantilla estructurada			
Nombre	<i>TemplateIdentifier</i> [( <i>TemplateFormalParList</i> )]		
Grupo	<i>[GroupReference]</i>		
Tipo/firma	<i>TypeIdentifier</i>   <i>SignatureIdentifier</i>		
Derivado de	<i>[TemplateRef]</i>		
Codificación	<i>[TabFreeText]</i>		
Comentarios	<i>[TabFreeText]</i>		
Nombre de elemento	Valor de elemento	Codificación de elemento	Comentarios
<i>FieldReference</i>	<i>FieldValueOrAttrib</i>	<i>[TabFreeText]</i>	<i>[TabFreeText]</i>
Comentarios detallados		<i>[TabFreeText]</i>	

Figura 14/Z.141 – Proforma de plantilla estructurada

#### 6.13.1 Correspondencia

Se establece una correspondencia entre la proforma de plantilla estructurada y un enunciado de definición de plantilla estructurada TTCN-3. Las definiciones de plantilla estructurada son siempre definiciones de plantilla que tienen en el cuerpo de la plantilla *FieldSpecList*. Los tipos correspondientes se definen en una proforma de tipo estructurado.

Se establece una correspondencia entre los campos **comentarios** y **comentarios detallados** y los atributos de visualización en el enunciado *WithStatement* asociado con la definición de plantilla estructurada. Se establece una correspondencia entre el campo **codificación** y un atributo de codificación en el enunciado *WithStatement* asociado con la definición de plantilla estructurada.

Se establece una correspondencia entre el campo **comentarios** del cuadro de elementos y los atributos de visualización definidos por la referencia de campo en el enunciado *WithStatement* asociado con la definición de plantilla estructurada. Se establece una correspondencia entre el campo **codificación de elemento** y los atributos de codificación definidos por la referencia de campo en el enunciado *WithStatement* asociado con la definición de plantilla estructurada.

```

1: module TTCN3ModuleId "{"
2:   template BaseTemplate [DerivedDef] " := " TemplateBody with "{"
3:     [EncodeAttribute ";"]
4:     [CommentsAttribute ";"]
5:     [FieldEncodeAttribute ";"]
6:     [FieldCommentsAttribute ";"]
7:     [DetailedCommentsAttribute ";"]
8:   }"
9: }"

```

Ejemplo:

Plantilla estructurada			
<b>Nombre</b>	MyStructuredTemplate11(integer para1, boolean para2)		
<b>Grupo</b>			
<b>Tipo/firma</b>	MyStructuredType		
<b>Derivado de</b>	MyStructuredTemplate1		
<b>Codificación</b>	BER		
<b>Comentarios</b>	ejemplo de plantilla estructurada		
Nombre de elemento	Valor de elemento	Codificación de elemento	Comentarios
field1	13		primer campo
field2	para2	PER	segundo campo
field3	para1		tercer campo
<b>Comentarios detallados</b>	comentarios detallados		

Se corresponde con:

```

1: module MyModule {
2:   template MyStructuredType MyStructuredTemplate11(integer para1,
3:                                                     boolean para2)
4:     modifies MyStructuredTemplate1 := {
5:       field1 := 13,
6:       field2 := para2,
7:       field3 := para1
8:     } with {
9:       encode "BER";
10:      display "comments := example structured template";
11:      display (field1) "comments := first field";
12:      encode (field2) "PER";
13:      display (field2) "comments := second field";
14:      display (field3) "comments := third field";
15:      display "detailed comments := detailed comments";
16:    }
17: }

```

## 6.14 Función

Función			
Nombre	<i>FunctionIdentifier</i> ( [ <i>FunctionFormalParList</i> ])		
Grupo	<i>[GroupReference]</i>		
Se ejecuta en	<i>[ComponentType]</i>		
Tipo de retorno	<i>[Type]</i>		
Comentarios	<i>[TabFreeText]</i>		
Nombre de definición local	Tipo	Valor inicial	Comentarios
<i>VarConstOrTimerIdentifier</i>	<i>TypeOrTimer</i>	<i>[Expression   ConstantExpression]</i>	<i>[TabFreeText]</i>
Comportamiento			
<i>FunctionStatement</i>   <b>external</b>			
Comentarios detallados	<i>[TabFreeText]</i>		

Figura 15/Z.141 – Proforma de función

### 6.14.1 Correspondencia

Se establece una correspondencia entre la proforma de función y un enunciado de definición de función TTCN-3 o una definición de función externa. Esto se traduce en tres partes.

La primera parte está formada por los campos de encabezamiento. Se establece una correspondencia entre los campos **comentarios** y **comentarios detallados** y los atributos de visualización en el enunciado *WithStatement* asociado con la definición de función.

La segunda parte está formada por constantes, variables y temporizadores locales definidos en la definición de función. Estas definiciones pueden aparecer en cualquier parte del cuerpo de la función en el lenguaje núcleo, pero en la proforma están separados del resto del cuerpo de la función y se visualizan en un cuadro distinto. Ha de respetarse el orden de las definiciones, ya que pueden ser interdependientes. En la columna **Tipo** se pondrá la palabra clave **timer** para todos los temporizadores, y el tipo de constante precedido de la palabra clave **const** para las constantes. El campo **comentarios** se transforma en atributos de visualización definidos por el identificador local en el enunciado *WithStatement* asociado con la definición de función.

La tercera parte está formada por el cuerpo de la función en el lenguaje núcleo TTCN-3, excepto las constantes, variables y temporizadores locales.

En el caso de una función externa, en comportamiento tan solo figurará la palabra clave **external**.

```

1: module TTCN3ModuleId "{"
2:   function FunctionIdentifier "(" [FunctionFormalParList] ")"
3:     [runs on ComponentType]
4:     [return Type] "{"
5:     var Type VarIdentifier [":=" Expression] ";"
6:     timer TimerIdentifier [":=" Expression] ";"
7:     const Type ConstIdentifier [":=" ConstantExpression] ";"
8:     {FunctionStatement}
9:   }" with "{"
10:    [CommentsAttribute] ";"
11:    [VarConstOrTimerCommentsAttribute] ";"
12:    [DetailedCommentsAttribute] ";"
13:  }"
14: }"

```

Ejemplo:

Función			
<b>Nombre</b>	MyFunction(integer para1)		
<b>Grupo</b>			
<b>Se ejecuta en</b>	MyComponentType		
<b>Tipo de retorno</b>	booleano		
<b>Comentarios</b>	ejemplo de definición de función		
Nombre de definición local	Tipo	Valor inicial	Comentarios
MyLocalVar	booleano	falso	variable local
MyLocalConst	const float	60	constante local
MyLocalTimer	temporizador	15 * MyLocalConst	temporizador local
Comportamiento			
<pre> if (para1 == 21) {     MyLocalVar := true; } if (MyLocalVar) {     MyLocalTimer.start;     MyLocalTimer.timeout; } return (MyLocalVar); </pre>			
<b>Comentarios detallados</b>	comentarios detallados		

Se corresponde con:

```

1: module MyModule {
2:     function MyFunction(in integer para1)
3:         runs on MyComponentType
4:         return boolean {
5:             var boolean MyLocalVar := false;
6:             const float MyLocalConst := 60;
7:             timer MyLocalTimer := 15 * MyLocalConst;
8:
9:             if (para1 == 21) {
10:                 MyLocalVar := true;
11:             }
12:             if (MyLocalVar) {
13:                 MyLocalTimer.start;
14:                 MyLocalTimer.timeout;
15:             }
16:             return (MyLocalVar);
17:         } with {
18:             display "comments := example function definition";
19:             display (MyLocalVar) "comments := local variable";
20:             display (MyLocalConst) "comments := local constant";
21:             display (MyLocalTimer) "comments := local timer";
22:             display "detailed comments := detailed comments";
23:         }
24: }

```

## 6.15 Altstep

Altstep			
Nombre	AltstepIdentifier([AltstepFormalParList])		
Grupo	[GroupReference]		
Propósito	[TabFreeText]		
Se ejecuta en	[ComponentType]		
Comentarios	[TabFreeText]		
Nombre de definición local	Tipo	Valor inicial	Comentarios
. VarConstOrTimerIdentifier .	. TypeOrTimer [ArrayDef] .	. [Expression   ConstantExpression] .	. [TabFreeText] .
Comportamiento			
. AltGuardList .			
Comentarios detallados	[TabFreeText]		

Figura 16/Z.141 – Proforma de altstep

### 6.15.1 Correspondencia

Se establece una correspondencia entre la proforma de altstep y un enunciado de definición de altstep en TTCN-3. Esto se traduce en tres partes.

La primera parte está formada por los campos de encabezamiento. Se establece una correspondencia entre los campos **propósito**, **comentarios** y **comentarios detallados** y los atributos de visualización en el enunciado *WithStatement* asociado con la definición de altstep.

La segunda parte está formada por las constantes, variables y temporizadores locales definidos en la definición de altstep. Estas definiciones pueden aparecer en cualquier lugar del cuerpo del altstep en lenguaje núcleo, pero en la proforma se separan del resto del cuerpo de altstep y se visualizan en un cuadro distinto. Ha de respetarse el orden de las definiciones, ya que pueden ser interdependientes. En la columna **tipo** se pondrá la palabra clave **timer** para todos los temporizadores, y el tipo constante precedido de la palabra clave **const** para las constantes. El campo **comentarios** se transforma en atributos de visualización definidos por el identificador local en el enunciado *WithStatement* asociado con la definición de altstep.

La tercera parte está formada por la *AltGuardList* del altstep en lenguaje núcleo TTCN-3.

```

1: module TTCN3ModuleId "{"
2:   teststep AltstepIdentifier "(" [AltstepFormalParList] ")"
3:   [runs on ComponentType] "{"
4:     AltGuardList
5:   }" with "{"
6:     [PurposeAttribute ";"]
7:     [CommentsAttribute ";"]
8:     [VarConstOrTimerCommentsAttribute ";"]
9:     [DetailedCommentsAttribute ";"]
10:  }"
11: }"

```

Ejemplo:

Altstep			
<b>Nombre</b>	MyAltstep(integer para1)		
<b>Grupo</b>			
<b>Se ejecuta en</b>	MyComponentType		
<b>Propósito</b>	hacer algo		
<b>Comentarios</b>	ejemplo de definición de altstep		
Nombre de definición local	Tipo	Valor inicial	Comentarios
MyLocalVar	Booleano	falso	variable local
MyLocalConst	const float	60	constante local
MyLocalTimer	temporizador	15 * MyLocalConst	temporizador local
Comportamiento			
<pre>[] PC01.receive(MyTemplate(para1, CompVar)) {     verdict.set(inconc); } [] PC02.receive {     repeat; } [] CompTimer.timeout {     verdict.set(fail);     stop; }</pre>			
<b>Comentarios detallados</b>	comentarios detallados		

Se corresponde con:

```

1: module MyModule {
2:     altstep MyTeststep(integer para1) runs on MyComponentType {
3:         var boolean MyLocalVar := false;
4:         const float MyLocalConst := 60;
5:         timer MyLocalTimer := 15 * MyLocalConst;
6:
7:         [] PC01.receive(MyTemplate(para1, CompVar)) {
8:             verdict.set(inconc);
9:         }
10:        [] PC02.receive {
11:            repeat;
12:        }
13:        [] CompTimer.timeout {
14:            verdict.set(fail);
15:            stop;
16:        }
17:    } with {
18:        display "purpose := to do something";
19:        display "comments := example altstep definition";
20:        display (MyLocalVar) "comments := local variable";
21:        display (MyLocalConst) "comments := local constant";
22:        display (MyLocalTimer) "comments := local timer";
23:        display "detailed comments := detailed comments";
24:    }
25: }
```

## 6.16 Caso de prueba

Caso de prueba			
Nombre	TestcaseIdentifier( [TestcaseFormalParList] )		
Grupo	[GroupReference]		
Propósito	[TabFreeText]		
Interfaz del sistema	[ComponentType]		
Tipo de MTC	ComponentType		
Comentarios	[TabFreeText]		
Nombre de definición local	Tipo	Valor inicial	Comentarios
. VarConstOrTimerIdentifier .	. TypeOrTimer .	. [Expression   ConstantExpression] .	. [TabFreeText] .
Comportamiento			
. FunctionStatement .			
Comentarios detallados	[TabFreeText]		

Figura 17/Z.141 – Proforma de caso de prueba

### 6.16.1 Correspondencia

Se establece una correspondencia entre la proforma de caso de prueba y un enunciado de definición de caso de prueba en TTCN-3. Esto se traduce en tres partes.

La primera parte está formada por los campos de encabezamiento. Se establece una correspondencia entre los campos **propósito**, **comentarios** y **comentarios detallados** y los atributos de visualización en el enunciado *WithStatement* de la definición de caso de prueba.

La segunda parte está formada por las constantes, variables y temporizadores locales definidos en la definición de caso de prueba. Estas definiciones pueden aparecer en cualquier parte del cuerpo del caso de prueba en lenguaje núcleo, pero en la proforma se separan del resto del cuerpo de caso de prueba y se visualizan en un cuadro distinto. Ha de respetarse el orden de las definiciones, ya que pueden ser interdependientes. En la columna **tipo** se pondrá la palabra clave **timer** para todos los temporizadores, y el tipo de constante precedido de la palabra clave **const** para las constantes. El campo **comentarios** se transforma en atributos de visualización definidos por el identificador local en el enunciado *WithStatement* asociado con la definición de caso de prueba.

La tercera parte consta del cuerpo del caso de prueba en lenguaje núcleo TTCN-3 excepto las constantes, variables y temporizadores locales.

```

1:  module TTCN3ModuleId "{"
2:      testcase TestcaseIdentifier [TestcaseFormalParList]
3:          [runs on ComponentType]
4:          [system ComponentType] "{"
5:          var Type VarIdentifier [":=" Expression] ";"
6:          timer TimerIdentifier [":=" Expression] ";"
7:          const Type ConstIdentifier ":=" ConstantExpression;
8:          {FunctionStatement}
9:      "}" with "{"
10:         [CommentsAttribute] ";"
11:         [PurposeAttribute] ";"
12:         [VarConstOrTimerCommentsAttribute] ";"
13:         [DetailedCommentsAttribute] ";"
14:     "}"
15: "}"

```

Ejemplo:

Altstep			
<b>Nombre</b>	MyTestcase(integer para1)		
<b>Grupo</b>			
<b>Propósito</b>	hacer algo útil		
<b>Interfaz del sistema</b>	MyComponentType		
<b>Tipo de MTC</b>	MyComponentType		
<b>Comentarios</b>	ejemplo de definición de caso de prueba		
Nombre de definición local	Tipo	Valor inicial	Comentarios
MyLocalVar	Booleano	falso	variable local
MyLocalConst	const float	60	constante local
MyLocalTimer	temporizador	15 * MyLocalConst	temporizador local
Comportamiento			
<pre> default.activate { [expand] OtherwiseFail(); }; /* Default activation */ ISAP1.send(ICONreq {}); /* Inline template definition */ alt {   [] MSAP2.receive(Medium_Connection_Request()) { /* use of a template */     MSAP2.send(MDATreq Medium_Connection_Confirmation());     alt {       [] ISAP1.receive(ICONconf {}); {         ISAP1.send(Data_Request(TestSuitePar) );         alt {           [] MSAP2.receive(Medium_Data_Transfer()) {             MSAP2.send(MDATreq cmi_synch1());             ISAP1.send(IDISreq {});           }           [] ISAP1.receive(IDISind {}) {             verdict.set(inconclusive);             stop();           }         }       }     }   }   [] MSAP2.receive(MDATind_Connection_Request()) {     verdict.set(inconclusive);     stop();   }   [] ISAP1.receive(IDISind {}) {     verdict.set(inconclusive);     stop();   } } [] ISAP1.receive(IDISind {}) {   verdict.set(inconclusive);   stop(); } </pre>			
<b>Comentarios detallados</b>	comentarios detallados		



Se corresponde con:

```
1: module MyModule {
2:     testcase MyTestcase(in integer para1)
3:         runs on MyComponentType
4:         system MyComponentType {
5:             var boolean MyLocalVar := false;
6:             const float MyLocalConst := 60;
7:             timer MyLocalTimer := 15 * MyLocalConst;
8:             var default MyDefault := activate(OtherwiseFail());
9:
10:            ISAP1.send(ICONreq:{}); /* Inline template definition */
11:            alt {
12:                /* use of a template */
13:                [] MSAP2.receive(Medium_Connection_Request()) {
14:                    MSAP2.send(MDATreq:Medium_Connection_Confirmation());
15:                    alt {
16:                        [] ISAP1.receive(ICONconf:{}) {
17:                            ISAP1.send(Data_Request(TestSuitePar));
18:                            alt {
19:                                [] MSAP2.receive(Medium_Data_Transfer()) {
20:                                    MSAP2.send(MDATreq:cmi_synch1());
21:                                    ISAP1.send(IDISreq:{});
22:                                }
23:                                [] ISAP1.receive(IDISind:{}) {
24:                                    verdict.set(inconc);
25:                                    stop;
26:                                }
27:                            }
28:                        }
29:                        [] MSAP2.receive(MDATind_Connection_Request()) {
30:                            verdict.set(inconc);
31:                            stop;
32:                        }
33:                        [] ISAP1.receive(IDISind:{}) {
34:                            verdict.set(inconc);
35:                            stop;
36:                        }
37:                    }
38:                }
39:                [] ISAP1.receive(IDISind:{}) {
40:                    verdict.set(inconc);
41:                    stop;
42:                }
43:            }
44:        } with {
45:            display "purpose := do something useful";
46:            display "comments := example testcase definition";
47:            display (MyLocalVar) "comments := local variable";
48:            display (MyLocalConst) "comments := local constant";
49:            display (MyLocalTimer) "comments := local timer";
50:            display "detailed comments := detailed comments";
51:        }
52:    }
```

## 7 Producciones BNF

1. TabFreeText ::= [ExtendedAlphaNum]
2. GroupReference ::= {GroupIdentifier "/"}
3. EncRuleIdentifier ::= Identifier
4. CommentsAttribute ::= **display** "" "comments" " := " TabFreeText ""
5. DetailedCommentsAttribute ::= **display** "" "detailed comments" " := " TabFreeText ""
6. TTCN3ModuleId ::= ModuleIdentifier [ DefinitiveIdentifier ]

```

7.  ModuleAttributes ::= TabularPresentationFormatAttribute ";"
                                ModuleVersionAttribute ";"
                                ModuleDateAttribute ";"
                                ModuleBaseStandardRefAttribute ";"
                                ModuleTestStandardRefAttribute ";"
                                ModulePICSRefAttribute ";"
                                ModulePIXITRefAttribute ";"
                                ModuleTestMethodAttribute ";"
                                ModuleCommentsAttribute ";"
                                ModuleDetailedCommentsAttribute ";"
8.  TabularPresentationFormatAttribute ::=
    display "" "presentation format := ETSI Tabular version" MajorVersion
    "." MinorVersion ""
9.  MajorVersion ::= Number
10. MinorVersion ::= Number
11. ModuleVersionAttribute ::=
    display "" "module version" " := " TabFreeText ""
12. ModuleDateAttribute ::=
    display "" "module date" " := " TabFreeText ""
13. ModuleBaseStandardRefAttribute ::=
    display "" "module base standards ref" " := " TabFreeText ""
14. ModuleTestStandardRefAttribute ::=
    display "" "module test standards ref" " := " TabFreeText ""
15. ModulePICSRefAttribute ::=
    display "" "module pics ref" " := " TabFreeText ""
16. ModulePIXITRefAttribute ::=
    display "" "module pixit ref" " := " TabFreeText ""
17. ModuleTestMethodAttribute ::=
    display "" "module test method" " := " TabFreeText ""
18. ModuleCommentsAttribute ::=
    display "" "module comments" " := " TabFreeText ""
19. ModuleDetailedCommentsAttribute ::=
    display "" "module detailed comments" " := " TabFreeText ""
20. ModuleParPicsPixitRefAttribute ::=
    display "(" ModuleParIdentifier ")"
    "" "pics/pixit ref" " := " TabFreeText ""
21. ModuleParComments ::=
    display "(" ModuleParIdentifier ")"
    "" "comments" " := " TabFreeText ""
22. ImportsSourceRefAttribute ::=
    display "" "imports source ref" " := " TabFreeText ""
23. ImportsSourceDefinitionCommentsAttribute ::=
    display "(" ImportIdentifier ")"
    "" "comments" " := " TabFreeText ""
24. ImportSpecification ::= ( (Identifier | FullGroupIdentifier) | AllKeyword )
    [ ExceptionsDef ]
    /* STATIC SEMANTIC: FullGroupIdentifier shall only be used for group
    imports. */
25. EncodeAttribute ::= encode "" TabFreeText ""
26. SimpleTypesDetailedCommentsAttribute ::=
    display "" "simple types detailed comments" " := " TabFreeText ""
27. StructureType ::= record | union | set
28. FieldCommentsAttribute ::=
    display "(" FieldIdentifier ")" "" "comments" " := " TabFreeText ""
29. FieldEncodeAttribute ::=
    encode "(" FieldIdentifier ")" "" TabFreeText ""
30. SequenceOfTypesDetailedCommentsAttribute ::=
    display "" "sequence of types detailed comments" " := " TabFreeText ""
31. NamedValueCommentsAttribute ::=
    display "(" NamedValueIdentifier ")"
    "" "comments" " := " TabFreeText ""
32. TypeOrSignatureCommentsAttribute ::=
    display "(" TypeOrSignatureIdentifier ")"
    "" "comments" " := " TabFreeText ""

```

```

33. PortCommentsAttribute ::=
    display "(" PortIdentifier ")"
    "" "comments" " := " TabFreeText ""
34. ConstantsDetailedCommentsAttribute ::=
    display "" "simple types detailed comments" " := " TabFreeText ""
35. ExceptionCommentsAttribute ::=
    display "(" Type ")"
    "" "comments" " := " TabFreeText ""
36. VarConstOrTimerCommentsAttribute ::=
    display "(" VarConstOrTimerIdentifier ")"
    "" "comments" " := " TabFreeText ""
37. PurposeAttribute ::= display "" "purpose" " := " TabFreeText ""
38. SimpleTemplatesDetailedCommentsAttribute ::=
    display "" "simple templates detailed comments" " := " TabFreeText ""

```





## 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
<b>Serie Z</b>	<b>Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación</b>