



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

Z.141

(07/2001)

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

Técnicas de descripción formal

**Notación combinada arborescente y tabular
versión 3: Formato de presentación tabular**

Recomendación UIT-T Z.141

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES UIT-T DE LA SERIE Z
**LENGUAJES Y ASPECTOS GENERALES DE SOPORTE LÓGICO PARA SISTEMAS DE
TELECOMUNICACIÓN**

TÉCNICAS DE DESCRIPCIÓN FORMAL	
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
LENGUAJES DE PROGRAMACIÓN	
CHILL: el lenguaje de programación del UIT-T	Z.200–Z.209
LENGUAJE HOMBRE-MÁQUINA	
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.399
CALIDAD DE SOPORTES LÓGICOS DE TELECOMUNICACIONES	Z.400–Z.499
MÉTODOS PARA VALIDACIÓN Y PRUEBAS	Z.500–Z.599

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

Recomendación UIT-T Z.141

Notación combinada arborescente y tabular versión 3: Formato de presentación tabular

Resumen

Esta Recomendación define el formato de presentación tabular de la notación combinada arborescente y tabular versión 3 (TTCN-3).

Orígenes

La Recomendación UIT-T Z.141, preparada por la Comisión de Estudio 10 (2001-2004) del UIT-T, fue aprobada por el procedimiento de la Resolución 1 de la AMNT el 22 de julio 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 2001

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

ÍNDICE

Página

1	Ámbito	1
2	Referencias.....	1
3	Abreviaturas.....	1
4	Introducción	1
5	Convenios	2
5.1	Metanotación sintáctica	2
5.2	Proformas.....	3
6	Reglas generales de correspondencia	3
7	Proformas.....	4
7.1	Control de sucesión de pruebas	4
7.1.1	Correspondencia	4
7.2	Parámetros de sucesión de pruebas.....	6
7.2.1	Correspondencia	6
7.3	Importación de módulos	7
7.3.1	Correspondencia	7
7.4	Codificación.....	8
7.4.1	Correspondencia	8
7.5	Tipos simples	9
7.5.1	Correspondencia	9
7.6	Tipos estructurados	10
7.6.1	Correspondencia	10
7.7	Tipos de puertos.....	11
7.7.1	Correspondencia	11
7.8	Tipos de componentes	12
7.8.1	Correspondencia	12
7.9	Constantes.....	13
7.9.1	Correspondencia	13
7.10	Firma.....	14
7.10.1	Correspondencia	14
7.11	Plantillas simples	15
7.11.1	Correspondencia	15
7.12	Plantilla estructurada.....	16
7.12.1	Correspondencia	16
7.13	Función	17
7.13.1	Correspondencia	17

	Página
7.14	Valores por defecto 19
7.14.1	Correspondencia 19
7.15	Denominación alternativa 20
7.15.1	Correspondencia 20
7.16	Caso de prueba..... 21
7.16.1	Correspondencia 21
8	Tabular presentationformat BNF (BNF del formato de presentación tabular) 23
8.1	ReferenceProforma (Proforma de referencia)..... 24
8.2	ParametersProforma (Proforma de parámetros) 24
8.3	ControlProforma (Proforma de control) 25
8.4	ImportsProforma (Proforma de importaciones)..... 25
8.5	EncodingProforma (Proforma de codificación)..... 25
8.6	SimpleTypesProforma (Proforma de tipos simples)..... 26
8.7	StructuredTypesProforma (Proforma de tipos estructurados) 26
8.8	PortTypeProforma (Proforma de tipo de puerto)..... 26
8.9	ComponentTypeProforma (Proforma de tipo de componente) 27
8.10	ConstantsProforma (Proforma de constantes) 27
8.11	SignatureProforma (Proforma de firma)..... 27
8.12	SimpleTemplatesProforma (Proforma de plantillas simples)..... 27
8.13	StructuredTemplatesProforma (Proforma de plantillas estructuradas)..... 28
8.14	FunctionProforma (Proforma de función) 28
8.15	DefaultsProforma (Proforma de valores por defecto)..... 29
8.16	NamedAltProforma (Proforma de denominación alternativa)..... 29
8.17	TestcaseProforma (Proforma de caso de prueba) 29

Recomendación UIT-T Z.141

Notación combinada arborescente y tabular versión 3: Formato de presentación tabular

1 Ámbito

Esta Recomendación define el formato de presentación tabular de la notación combinada arborescente y tabular versión 3 (o TTCN-3, *tree and tabular combined notation*). Se basa en el lenguaje núcleo TTCN-3 definido en UIT-T Z.140 [1].

La especificación de otros formatos queda fuera del ámbito de la presente 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.

- [1] UIT-T Z.140 (2001), *Notación combinada arborescente y tabular – Versión 3 – (TTCN-3): Lenguaje núcleo.*

3 Abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

ASN.1	Notación de sintaxis abstracta uno (<i>abstract syntax notation one</i>)
ATS	Sucesión de pruebas abstractas (<i>abstract test suite</i>)
BNF	Forma Backus-Naur (<i>Backus-Naur form</i>)
IUT	Implementación sometida a prueba (<i>implementation under test</i>)
MTC	Componente de prueba principal (<i>master test component</i>)
PICS	Enunciado de conformidad de implementación de protocolo (<i>protocol implementation conformance statement</i>)
PIXIT	Información suplementaria de implementación de protocolo para pruebas (<i>protocol implementation eXtra information for testing</i>)
TTCN	Notación combinada arborescente y tabular (<i>tree and tabular combined notation</i>)

4 Introducción

El formato de presentación tabular es un formato gráfico similar en apariencia y funcionalidad a versiones precedentes de TTCN, que están orientadas a la realización de pruebas de conformidad. El lenguaje núcleo de TTCN-3 se define en UIT-T Z.140 [1] y proporciona una sintaxis completa textual, semántica estática y semántica operacional, definiendo asimismo la utilización del lenguaje con ASN.1. El formato tabular proporciona una forma alternativa de visualizar el lenguaje núcleo, así como de enfatizar aquellos aspectos que son específicos de los requisitos de una sucesión de pruebas de conformidad normalizada.

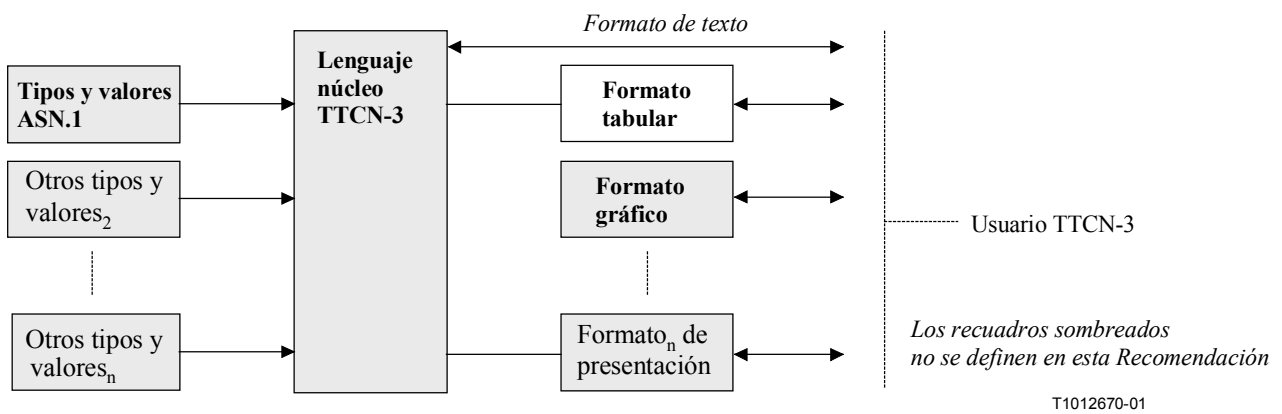


Figura 1/Z.141 – Visión de usuario del lenguaje núcleo y de los diversos formatos de presentación

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 que se han utilizado en la definición de las proformas TTCN y en la gramática del 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
abc	un símbolo terminal abc
"abc"	un símbolo terminal abc

En la cláusula 8 de esta Recomendación se definen las posibles producciones de BNF. Las producciones que no se definen en la cláusula 8 pueden encontrarse en UIT-T Z.140 [1].

5.2 Proformas

- a) En cada cuadro de una sucesión de pruebas TTCN aparecerá literalmente texto en negrita (**como éste**).
- b) En una sucesión de pruebas TTCN no aparecerá literalmente texto en cursiva (*como éste*). 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.
- c) Un fondo en tono gris significa que el campo (o la fila, o la columna) es facultativo.

6 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 cada elemento sintáctico de cada proforma existe una transformación asociada.

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 transformaciones sería un campo de 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 del lenguaje en el formato de presentación. Estos elementos no tienen un significado directo en el lenguaje núcleo y se expresan utilizando el enunciado **with**.

Los enunciados **with** tienen un formato común con la forma siguiente:

```
with display "<ProfomaIdentifier> { <ElementIdentifier1> := FreeText;  
                                <ElementIdentifier2> := FreeText;  
                                <ElementIdentifierN> := FreeText }"
```

<ProfomaIdentifier> es el nombre del cuadro asociado en el formato tabular y *<ElementIdentifier>* es el nombre del campo o elemento de dicho cuadro que se define en el enunciado. El valor del campo o elemento se especifica utilizando texto libre a continuación del carácter "=". En esta Recomendación se realiza la definición BNF precisa para la interpretación de la cadena de visualización.

La sintaxis y la semántica que se especifica en esta Recomendación son propias del formato de presentación tabular del ETSI. 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:

```
module ModuleName()  
{  
}  
with { display "PresentationFormatIdentifier" }  
  
PresentationFormatIdentifier ::= PresentationKeyword FormatKeyword  
AssignmentChar "ETSI Tabular v1.0"  
PresentationKeyword ::= presentation  
FormatKeyword ::= format
```

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

7 Proformas

7.1 Control de sucesión de pruebas

Control de sucesión de pruebas			
Nombre	:	<i>TTCN3ModuleId</i>	
Versión	:	<i>VersionIdentifier</i>	
Fecha	:	<i>Texto libre</i>	
Ref. normalizada de base	:	<i>Texto libre</i>	
Ref. normalizada de prueba	:	<i>Texto libre</i>	
Ref. PICS	:	<i>Texto libre</i>	
Ref. PIXIT	:	<i>Texto libre</i>	
Método(s) de prueba	:	<i>Texto libre</i>	
Comentarios detallados	:	<i>[Texto libre]</i>	
Nombre	Tipo	Valor inicial	Comentarios
<i>[VarConstOrTimerIdentifier]</i>	<i>[TypeOrTimer]</i>	<i>[ConstantExpression]</i>	<i>[Texto libre]</i>
.	.	.	.
.	.	.	.
Comportamiento			Comentarios
.			.
<i>ModuleControlBody</i>			<i>[Texto libre]</i>
.			.
.			.
Comentarios detallados: <i>[Texto libre]</i>			

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

7.1.1 Correspondencia

La proforma del control de sucesión de pruebas se traduce en dos partes. La primera parte es la parte de control del módulo de lenguaje núcleo TTCN-3. Los campos de comentarios se convierten en un enunciado **with** asociado con la definición de control en el lenguaje núcleo. La información del encabezamiento de la proforma del control de sucesión de pruebas se convierte en un enunciado **with** asociado con el módulo TTCN-3 en su conjunto.

```

module MyModule()
{
  control{
    var Type VarIdentifier [ "==" ConstantExpression]
    timer TimerIdentifier [ "==" ConstantExpression]
    const Type ConstIdentifier "==" ConstantExpression

    ModuleControlBody

  } with display "ControlProforma"
}
with display "ReferenceProforma"

```

Ejemplo:

Control de sucesión de pruebas			
Nombre	:	MyATS	
Versión	:	1.1	
Fecha	:	23 de mayo de 1999	
Ref. de normas de base	:		
Ref. de normas de prueba	:		
Ref. PICS	:		
Ref. PIXIT	:		
Método(s) de prueba	:	local	
Comentarios detallados	:	ATS escrito por STF 133	
Nombre		Tipo	Valor inicial
X		integer	7
T1		timer	15 min
Comportamiento			Comentarios
<pre> /* group1/ */ /* group1_1/ */ execute(test1); execute(test2); /* group1_2/ */ execute(test3); execute(test4); /* group2/ */ execute(test5); </pre>			basic tests check capability 1
Comentarios detallados:			

Se corresponde con:

```

module MyATS()
{
  control{
    var integer x := 7;
    timer T1 := 15 min;

    /* group1/ */
      /* group1_1/ */
        execute( test1);
        execute( test2);
      /* group1_2/ */
        execute( test3);
        execute( test4);
    /* group2/ */
      execute( test5);
  } with display " Control { extracomments := "basic tests
    check capability 1"}";

} with
{
display "presentation format := "ETSI Tabular v1.0";
display "reference { version := "1.1";
                    date := "23 May 1999";
                    testmethod := "local";
                    detailedcomments := "ATS written by STF 133" }"
}

```

7.2 Parámetros de sucesión de pruebas

Parámetros de sucesión de pruebas				
Grupo : <i>[GroupReference]</i>				
Nombre	Tipo	Valor inicial	Ref. PICS/PIXIT	Comentarios
<i>ModuleParIdentifier</i>	<i>ModuleParType</i>	<i>[ConstantExpression]</i>	<i>[Texto libre]</i>	<i>[Texto libre]</i>
Comentarios detallados: <i>[Texto libre]</i>				

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

7.2.1 Correspondencia

Se establece una correspondencia entre todas las entradas del cuadro de parámetros y la lista de parámetros del módulo TTCN-3 asociado. Las referencias PICS/PIXIT y los comentarios se hacen corresponder con los enunciados **with** del módulo TTCN-3.

```

module MyModule(ModuleType ModuleParIdentifier)
{
}
with display "ParametersProforma";

```

Ejemplo:

Parámetros de sucesión de pruebas				
Grupo : PICS/				
Nombre	Tipo	Valor inicial	Ref. PICS/PIXIT	Comentarios
CAP_1	boolean	true	A.1.3	Option 1 implemented by IUT
Comentarios detallados:				

Se corresponde con:

```

module MyModule(boolean CAP_1 := true)
{
}
with display "parameters {
  group := PICS/;
  pics pixet := { CAP_1 := "A.1.3"}
  comments := { CAP1 := "Option 1 implemented by IUT"}
}"

```

7.3 Importación de módulos

Importaciones			
Nombre de fuente	:	<i>ModuleIdentifier [DefinitiveIdentifier] ModuleIdentifier [DefinitiveIdentifier]</i>	
Lenguaje fuente	:	<i>[LanguageSpec]</i>	
Grupo	:	<i>[GroupReference]</i>	
Ref. de fuente	:	<i>[Texto libre]</i>	
Codificación	:	<i>[Texto libre]</i>	
Comentarios	:	<i>[Texto libre]</i>	
Tipo	Nombre	NR	Comentarios
.	.	.	.
<i>ImportType</i>	<i>ImportIdentifier</i>	<i>Mark</i>	<i>[Texto libre]</i>
.	.	.	.
.	.	.	.
Comentarios detallados: <i>[Texto libre]</i>			

Figura 4/Z.141 – Proforma de importaciones

7.3.1 Correspondencia

Se establece una correspondencia entre la proforma de importaciones y el grupo de enunciados de importación en el lenguaje núcleo TTCN-3. El nombre de fuente, el tipo de importación, el nombre de importación y los elementos tabulares de recursión se utilizan directamente en el correspondiente enunciado de importación del lenguaje núcleo. El grupo recibe el nombre de `Imports`, añadiendo un número único al final del identificador cuando sea necesario para que el nombre del grupo quede inequívocamente determinado. Todos los campos restantes se traducen a un enunciado `with` asociado con el grupo en el que están incluidos.

```

module MyModule()
{
  group Imports1 {
    import ImportType ImportIdentifier from
ModuleIdentifier[DefinitiveIdentifier] [language LanguageIdentifier];
  }
with { display "ImportsProforma";
        encode "FreeText" }

```

Ejemplo:

Importaciones			
Nombre de fuente	:	ModuleA	
Ref. de fuente	:	EN 800 900 version 2	
Codificación	:		
Comentarios	:	declaraciones de importación desde un ATS existente	
Tipo	Nombre	NR	Comentarios
todos constantes			
tipo	MyType	*	(1)
grupo	AtoU_CTR		
Comentarios detallados: (1) Tick indicates: import recursively what is needed for MyType definition.			

Se corresponde con:

```

module MyModule()
{
  group Imports1 {
    import all constant from ModuleA;
    import type MyType from ModuleA;
    import group AtoU_CTR from ModuleA;
  }
  with display "imports" { source := "EN 800 900 version 2";
                           comments := "importing declarations from an existing
                           ATS";
                           extracomments := "(1)";
                           detailedcomments := "(1) asterisk indicates: import
                           recursively what is needed for
                           MyType definition}"
}

```

7.4 Codificación

Definiciones de codificación			
Grupo	:	[GroupReference]	
Nombre	Referencia	Valor por defecto	Comentarios
.	.	.	.
<i>EncodingRuleIdentifier</i>	<i>Texto libre</i>	<i>[BooleanExpression]</i>	<i>[Texto libre]</i>
.	.	.	.
.	.	.	.
Comentarios detallados: <i>[Texto libre]</i>			

Figura 5/Z.141 – Proforma de definiciones de codificación

7.4.1 Correspondencia

Se establece una correspondencia entre la proforma de codificación y una serie de enunciados en el enunciado **with** asociado con el módulo núcleo de TTCN-3. Todos los elementos del cuadro se hacen corresponder con enunciados de visualización. Además, se añade un enunciado de codificación al enunciado **with** para la regla de codificación cuyo valor por defecto da como resultado **true** (verdadero).

```

module MyModule()
{
}
with {
  display " EncodingProforma ";
  encode "EncodingRuleIdentifier"
}

```

Ejemplo:

Definiciones de codificación			
Nombre de regla de codificación	Referencia	Valor por defecto	Comentarios
BER	ISO/IEC 8825-1: 1993	TRUE	Basic Encoding Rules
PER	ISO/IEC 8825-1: 1993		Packed Encoding Rules
DER	ISO/IEC 8825-1: 1993		Distinguished Encoding Rules
Comentarios detallados:			

Se corresponde con:

```

module MyModule()
{
}
with {
display "encoding {      reference := {      BER := "ISO/IEC 8825-1: 1993",
                                         PER := "ISO/IEC 8825-1: 1993",
                                         DER := "ISO/IEC 8825-1: 1993"};
                                default  := {      BER := TRUE};
                                comments := {      BER := "Basic Encoding Rules",
                                         PER := "Packed Encoding Rules",
                                         DER := "Distinguished Encoding Rules"}";
encode "BER" }

```

7.5 Tipos simples

Tipos simples			
Grupo	:	[GroupReference]	
Nombre	Definición	Codificación	Comentarios
<i>SubTypeIdentifier</i>	<i>Type [SubTypeSpec]</i>	<i>[Texto libre]</i>	<i>[Texto libre]</i>
Comentarios detallados: <i>[Texto libre]</i>			

Figura 6/Z.141 – Proforma de tipos simples

7.5.1 Correspondencia

Se establece una correspondencia entre la proforma de tipos simples y un grupo TTCN-3 que contiene una serie de enunciados de definición de tipo. La referencia del grupo y los comentarios detallados se hacen corresponder con enunciados de visualización en el enunciado **with** asociado al grupo. Los campos de codificación y de comentarios se hacen corresponder con enunciados con el enunciado **with** asociado a las distintas definiciones de tipos.

El grupo se denomina SimpleTypes_n, donde "n" es un número entero que se utiliza para distinguir más de un grupo de tipo simple.

```

module MyModule()
{
  group SimpleTypes1 {
    type Type SubTypeIdentifier SubTypeSpec
  }
  with {
    encode (SubTypeIdentifier) "FreeText";
    display "SimpleTypesProforma ";
  }
}

```

Ejemplo:

Tipos simples			
Nombre	Definición	Codificación	Comentarios
EQ_NUMBER	integer (1 .. 20)		
Comentarios detallados:			

Se corresponde con:

```

module MyModule()
{
  group SimpleTypes1 {
    type integer EQ_NUMBER (1..20)
  }
  with display "simpletypes {}";
}

```

7.6 Tipos estructurados

Tipo estructurado			
Nombre	:	<i>StructTypeIdentifier</i>	
Grupo	:	<i>[GroupReference]</i>	
Estructura	:	<i>StructureType</i>	
Codificación	:	<i>[Texto libre]</i>	
Comentarios	:	<i>[Texto libre]</i>	
Nombre de elemento	Definición de tipo	Codificación de campo	Comentarios
.	.	.	.
<i>StructFieldIdentifier</i>	<i>Type [SubTypeSpec] [OptionalKeyword]</i>	<i>[Texto libre]</i>	<i>[Texto libre]</i>
.	.	.	.
Comentarios detallados: <i>[Texto libre]</i>			

Figura 7/Z.141 – Proforma de tipo estructurado

7.6.1 Correspondencia

Se establece una correspondencia entre la proforma de tipo estructurado y el enunciado de definición de tipo en TTCN-3, haciéndose corresponder los campos grupo y comentarios con los enunciados de visualización en el correspondiente enunciado **with**.

```

module MyModule()
{
  type StructureType StructTypeIdentifier
  {
    Type FieldIdentifier [ SubtypeSpec][ OptionalKeyword]
  }
  with {
    display "StructuredTypeProforma";
    encode "FreeText";
    encode (StructFieldIdentifier) "FreeText";
  }
}

```


Ejemplo:

Tipo estructurado			
Nombre	:	<i>MaleMind</i>	
Grupo	:		
Estructura	:	<i>record</i>	
Codificación	:		
Comentarios	:		
Nombre de elemento	Definición de tipo	Codificación de campo	Comentarios
Car	integer		
Money	integer		
Football	octetstring		
Comentarios detallados:			

Se corresponde con:

```

module MyModule()
{
    type record MaleMind
    {
        integer      Car,
        integer      Money,
        octetstring  Football
    }
    with display "structuredtype {
        comments := "";
        comments := {}
        detailedcomments := ""
    }
}

```

7.7 Tipos de puertos

Tipo de puerto	
Nombre	: <i>PortTypeIdentifier</i>
Grupo	: <i>[GroupReference]</i>
Modelo de Comunicación	: <i>PortModelType</i>
Comentarios	: <i>[Texto libre]</i>
Definición de tipo	Comentarios
<i>PortTypeDef</i>	.
.	.
..	<i>[Texto libre]</i>
.	.
Comentarios detallados: <i>[Texto libre]</i>	

Figura 8/Z.141 – Proforma de tipo de puerto

7.7.1 Correspondencia

Se establece una correspondencia entre la proforma de tipo de puerto y una definición de tipo de puerto en TTCN-3, haciéndose corresponder los campos grupo y comentario con los enunciados de visualización en el correspondiente enunciado **with**.

```

module MyModule()
{
    type port PortTypeIdentifier PortModelType
    {
        PortTypeDef
    }
    with display "PortTypeProforma";
}

```

Ejemplo:

Tipo de puerto	
Nombre	: <i>MyPortType</i>
Grupo	:
Modelo de comunicación	: <i>message</i>
Comentarios	:
Definición de tipo	
in	<i>MsgType1, MsgType2;</i>
out	<i>MsgType3;</i>
Comentarios detallados:	

Se corresponde con:

```

module MyModule()
{
    type port MyPortType message
    {
        in MsgType1, MsgType2;
        out MsgType3;
    }
    with display "porttype { }";
}

```

7.8 Tipos de componentes

Tipo de componente			
Nombre	:	<i>ComponentTypeIdentifier</i>	
Grupo	:	<i>[GroupReference]</i>	
Comentarios	:	<i>[Texto libre]</i>	
Nombre	Tipo	Valor inicial	Comentarios
<i>[VarConstOrTimerIdentifier]</i>	<i>[TypeOrTimer]</i>	<i>[ConstantExpression]</i>	<i>[Texto libre]</i>
.	.	.	.
.	.	.	.
Definiciones de puerto			Comentarios
<i>PortList</i>			<i>[Texto libre].</i>
.			.
.			.
.			.
Comentarios detallados: <i>[Texto libre]</i>			

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

7.8.1 Correspondencia

Se establece una correspondencia entre la proforma de tipo de componente y la definición de tipo de componente en TTCN-3, haciéndose corresponder los campos grupo y comentario con los enunciados de visualización en el correspondiente enunciado **with**.

```

module MyModule()
{
    type component ComponentTypeIdentifier
    {
        var Type VarIdentifier [ "==" ConstantExpression]
        timer TimerIdentifier [ "==" ConstantExpression]
        const Type ConstIdentifier "==" ConstantExpression
        PortList
    }
    with display "ComponentTypeProforma";
}

```

Ejemplo:

Tipo de componente			
Nombre	:	MyComponentType	
Grupo	:		
Comentarios	:		
Nombre	Tipo	Valor inicial	Comentarios
x	Integer	7	
T1	timer	15 min	
Definiciones de puerto			Comentarios
MyMessagePortType PCO1, PCO2;			
MyOtherPortType PCO3, PCO4;			
Comentarios detallados:			

Se corresponde con:

```

module MyModule()
{
    type component MyComponentType
    {
        var integer x := 7;
        timer T1 := 15 min;

        MyMessagePortType PCO1, PCO2;
        MyOtherPortType PCO3, PCO4;
    }
    with display "componenttype {}";
}

```

7.9 Constantes

Constantes			
Grupo	:	<i>[GroupReference]</i>	
Nombre	Tipo	Valor	Comentarios
<i>ConstIdentifier</i>	<i>Type</i>	<i>ConstantExpression</i>	<i>[Texto libre]</i>
Comentarios detallados: <i>[Texto libre]</i>			

Figura 10/Z.141 – Proforma de constantes

7.9.1 Correspondencia

Se establece una correspondencia entre la proforma de constantes y un grupo que contiene declaraciones de constantes en el lenguaje TTCN-3. La referencia de grupo y los comentarios se hacen corresponder con enunciados de visualización en el enunciado **with** asociado. El grupo se denomina Constants_n, donde "n" es un número entero unívoco que se añade al final del identificador.

```

module MyModule()
{
    group Constants1 {
        const Type ConstIdentifier := ConstantExpression
    }
    with display "ConstantsProforma";
}

```

Ejemplo:

Constantes			
Grupo : Misc			
Nombre	Tipo	Valor	Comentarios
sel2 T1	boolean integer	(5 + TOTO) < 10 15	TOTO is a constant
Comentarios detallados:			

Se corresponde con:

```

module MyModule()
{
    group Constants1 {
        const boolean sel2 := (5 + TOTO) < 10;
        const integer T1 := 15;
    }
    with display "constants" {
        group := "Misc";
        comments := {sel2 ::= "TOTO is a
constant"}";
    }
}

```

7.10 Firma

Definición de firma	
Nombre :	<i>SignatureIdentifier&ParList</i>
Grupo :	<i>[GroupReference]</i>
Tipo de retorno :	<i>[Type]</i>
Comentarios :	<i>[Texto libre]</i>
Lista de excepciones	Comentarios
<i>ExceptionTypeList</i>	. . <i>[Texto libre]</i> .
Comentarios detallados: <i>[Texto libre]</i>	

Figura 11/Z.141 – Proforma de definición de firma

7.10.1 Correspondencia

Se establece una correspondencia entre la proforma de firma y una definición de firma en TTCN-3, haciéndose corresponder los campos grupo y comentario con los enunciados de visualización en el correspondiente enunciado **with**.

```

signature SignatureIdentifier&ParList [return Type]
{
    ExceptionTypeList
}
with display "SignatureProforma";

```

Ejemplo:

Definición de firma	
Nombre	: MySignature(in integer Par1, out float Par2)
Grupo	:
Tipo de retorno	: boolean
Comentarios	:
Lista de excepciones	Comentarios
integer, boolean, MyType	
Comentarios detallados:	

Se corresponde con:

```
signature MySignature( in interger Par1, out float Par2) return boolean
{
    integer, boolean, MyType
}
with display "signature {}";
```

7.11 Plantillas simples

Plantillas simples				
Grupo	: [GroupReference]			
Nombre	Tipo	Valor	Codificación	Comentarios
TemplateIdentifier	SimpleType	SingleValueOrAttrib	[Texto libre]	[Texto libre]
Comentarios detallados: [Texto libre]				

Figura 12/Z.141 – Proforma de plantillas simples

7.11.1 Correspondencia

Se establece una correspondencia entre la proforma de plantillas simples y un grupo TTCN-3 que contiene una serie de enunciados de definición de plantilla. La referencia de grupo y los comentarios detallados se hacen corresponder con enunciados de visualización en el enunciado **with** asociado al grupo. Los campos codificación y comentarios se hacen corresponder con enunciados con el enunciado **with** asociado a las distintas definiciones de plantillas.

El grupo se denomina SimpleTemplates_n, donde "n" es un número entero utilizado para distinguir más de un grupo de plantillas simples.

```
module MyModule()
{
    group SimpleTemplates1 {
        template SimpleType TemplateIdentifier := SingleVlaueOrAttrib
    }
    with {
        encode (TemplateIdentifier) "FreeText";
        display "SimpleTemplatesProforma ";
    }
}
```

Ejemplo:

Plantilla simple				
Nombre	Tipo	Valor	Codificación	Comentarios
AgeField	integer	?		
Comentarios detallados:				

Se corresponde con:

```

module MyModule()
{
  group SimpleTemplates1 {
    template integer AgeField := ?;
  }
  with display "simpletemplates {}";
}

```

7.12 Plantilla estructurada

Plantilla estructurada			
Nombre	:	<i>TemplateIdentifier&ParList</i>	
Grupo	:	<i>[GroupReference]</i>	
Tipo	:	<i>TemplateStructIdentifier</i>	
Codificación	:	<i>[Texto libre]</i>	
Comentarios	:	<i>[Texto libre]</i>	
Nombre de elemento	Valor del elemento	Codificación del elemento	Comentarios
.	.	.	.
<i>FieldReference</i>	<i>FieldValueOrAttrib</i>	.	<i>[Texto libre]</i>
.	.	<i>[Texto libre]</i>	.
.	.	.	.
Comentarios detallados: <i>[Texto libre]</i>			

Figura 13/Z.141 – Proforma de plantilla estructurada

7.12.1 Correspondencia

Se establece una correspondencia entre la proforma de plantilla estructurada y un enunciado de definición de plantilla en TTCN-3, haciéndose corresponder los campos grupo y comentario con los enunciados de visualización en el correspondiente enunciado **with**.

```

template [Type | Signature] TemplateIdentifier&ParList [modifies TemplateRef] :=
{
  FieldReference ::= FieldValueOrAttrib
}
with {
  display "StructutedTemplateProforma";
  encode "FreeText";
  encode (FieldReference) "FreeText";
}

```

Ejemplo:

Plantilla estructurada			
Nombre	:	Setup01	
Grupo	:		
Tipo	:	SetupMsgType	
Trayecto de derivación	:		
Codificación	:		
Comentarios	:		
Nombre de elemento	Valor de elemento	Codificación de elemento	Comentarios
MsgId	34		
CrLength	1		
CrValue	42		
IE1	?		
IE2	?		
Comentarios detallados:			

Se corresponde con:

```

template SetupMsgType Setup01 :=
{
    MsgId :=      34,
    CrLength :=   1,
    CrValue  :=   42,
    IE1 :=       ?,
    IE2 :=       ?
}
with display "structuredtemplate {}";

```

7.13 Función

Función			
Nombre	:	<i>FunctionIdentifier&ParList</i>	
Grupo	:	<i>[GroupReference]</i>	
Se ejecuta en	:	<i>[ComponentType]</i>	
Tipo de retorno	:	<i>[Type]</i>	
Comentarios	:	<i>[Texto libre]</i>	
Nombre	Tipo	Valor inicial	Comentarios
<i>[VarConstOrTimerIdentifier]</i>	<i>[TypeOrTimer]</i>	<i>[ConstantExpression]</i>	<i>[Texto libre]</i>
.	.	.	.
.	.	.	.
Definición de función			Comentarios
<i>TabularBehaviour</i>			<i>[Texto libre].</i>
.			.
.			.
Comentarios detallados: <i>[Texto libre]</i>			

Figura 14/Z.141 –Proforma de función

7.13.1 Correspondencia

Se establece una correspondencia entre la proforma de función y una definición de función o de función externa en TTCN-3, haciéndose corresponder los campos grupo y comentario con enunciados de visualización en el correspondiente enunciado **with**. La palabra clave **external** que antecede al nombre de función indica que la función se debería hacer corresponder con una función externa.

```

[external] function FunctionIdentifier&ParList [return Type]
[Runs On ComponentType]
{
    var Type VarIdentifier [:= ConstantExpression] ;
    timer TimerIdentifier [:= ConstantExpression] ;
    const Type ConstIdentifier := ConstantExpression

    TabularBehaviour
}
with display "FunctionProforma";

```

Ejemplo:

Función				
Nombre	:	MyFunction(in integer Par1)		
Grupo	:			
Se ejecuta en	:			
Tipo de retorno	:	boolean		
Comentarios	:			
Nombre		Tipo	Valor inicial	Comentarios
MyLocalVar T1		boolean timer	false 15 min	
Definición de función				Comentarios
<pre> if(Par1 = 21) { MyLocalVar := true; } if(MyLocalVar) { T1.start; T1.timeout; } return(MyLocalVar); </pre>				
Comentarios detallados:				

Se corresponde con:

```

function MyFunction( in integer Par1) return boolean
{
    var boolean MyLocalVar := false;
    timer T1 := 15 min;

    if( Par1 = 21 ) {
        MyLocalVar := true;
    }
    if( MyLocalVar) {
        T1.start;
        T1.timeout;
    }
    return( MyLocalVar);
}
with display "function";

```


7.14 Valores por defecto

Definición de valor por defecto	
Nombre	: <i>NamedAltIdentifier&ParList</i>
Grupo	: <i>[GroupReference]</i>
Propósito	: <i>[Texto libre]</i>
Comentarios	: <i>[Texto libre]</i>
Comportamiento	
<i>AltGuardList</i>	<i>[Texto libre].</i>
.	.
..	.
Comentarios detallados:	<i>[Texto libre]</i>

Figura 15/Z.141 – Proforma de definición de valor por defecto

7.14.1 Correspondencia

Se establece una correspondencia entre la proforma de valor por defecto y una definición de denominación alternativa en el módulo TTCN-3. Los campos grupo y comentario se hacen corresponder con enunciados de visualización en el enunciado `with` asociado a la definición.

```
named alt NamedAltIdentifier&ParList{
    AltGuardList
}
with display "default {purpose := ""};";
```

Ejemplo:

Definición de valor por defecto	
Nombre	: Default1
Grupo	:
Propósito	:
Comentarios	:
Comportamiento	
[] PCO2.receive(DL_EST_IN); PCO2.send(DL_EST_CO);	
[] PCO2.receive(DL_EST_CO); // do nothing	
Comentarios detallados:	

Se corresponde con:

```
named alt Default1{
    [ ] PCO2.receive( DL_EST_IN){
        PCO2.send( DL_EST_CO)
    }
    [ ] PCO2.receive( DL_EST_CO);
}
with display "default{ purpose := ""};";
```

7.15 Denominación alternativa

Definición de denominación alternativa	
Nombre	: <i>NamedAltIdentifier&ParList</i>
Grupo	: <i>[GroupReference]</i>
Propósito	: <i>[Texto libre]</i>
Comentarios	: <i>[Texto libre]</i>
Comportamiento	Comentarios
<i>AltGuardList</i>	.
.	.
..	.
Comentarios detallados:	<i>[Texto libre]</i>

Figura 16/Z.141 – Proforma de definición de denominación alternativa

7.15.1 Correspondencia

Se establece una correspondencia entre la proforma de denominación alternativa y una definición de denominación alternativa en el módulo TTCN-3. Se hacen corresponder los campos grupo y comentario con enunciados de visualización en el enunciado `with` asociado a la definición.

```
named alt NamedAltIdentifier&ParList{
    AltGuardList
}
with display "NamedAltProforma ";
```

Ejemplo:

Definición de denominación alternativa	
Nombre	: TS01
Grupo	:
Propósito	:
Comentarios	:
Comportamiento	Comentarios
[] PCO2.receive(DL_EST_IN); PCO2.send(DL_EST_CO);	
[] PCO2.receive(DL_EST_CO); // do nothing	
Comentarios detallados:	

Se corresponde con:

```
named alt Default1{
    [ ] PCO2.receive( DL_EST_IN){
        PCO2.send( DL_EST_CO)}
    [ ] PCO2.receive( DL_EST_CO);
}
with display "namedalt{ purpose := "; }";
}
```

7.16 Caso de prueba

Definición de caso de prueba			
Nombre	:	<i>TestcaseIdentifier&ParList</i>	
Grupo	:	<i>[GroupReference]</i>	
Propósito	:	<i>[Texto libre]</i>	
Interfaz del sistema	:	<i>[ComponentType]</i>	
Tipo de MTC	:	<i>ComponentType</i>	
Comentarios	:	<i>[Texto libre]</i>	
Nombre	Tipo	Valor inicial	Comentarios
<i>[VarConstOrTimerIdentifier]</i>	<i>[TypeOrTimer]</i>	<i>[ConstantExpression]</i>	<i>[Texto libre]</i>
.	.	.	.
.	.	.	.
.	.	.	.
Comportamiento			Comentarios
<i>TabularBehaviour</i>			.
.			.
..			.
			.
			.
			.
Comentarios detallados: <i>[Texto libre]</i>			

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

7.16.1 Correspondencia

Se establece una correspondencia entre la proforma de caso de prueba y una definición de caso de prueba en TTCN-3, haciéndose corresponder los campos grupo y comentario con enunciados de visualización en el correspondiente enunciado **with**. La lista de parámetros sólo puede contener variables de sucesión de pruebas.

```

testcase TestcaseIdentifier&ParList
runs on ComponentType[system ComponentType]
{
    var Type VarIdentifier [":=" ConstantExpression] ;
    timer TimerIdentifier [":=" ConstantExpression] ;
    const Type ConstIdentifier ":=" ConstantExpression ;

    TabularBehaviour
}
with display "TestcaseProforma";

```

Ejemplo:

Definición de caso de prueba			
Nombre	:	MyTestcase	
Grupo	:		
Propósito	:	Primer ejemplo de caso de prueba (Testcase)	
Interfaz del sistema	:		
Tipo de MTC	:	MyComponentType	
Comentarios	:		
Nombre	Tipo	Valor inicial	Comentarios
MyLocalVar TimerT1	integer timer	0 15 min	
Comportamiento			Comentarios
<pre> default.activate { [expand] OtherwiseFail(); }; /* Activación por defecto */ ISAP1.send(ICONreq {}); /* Definición de plantilla Inline */ alt { [] MSAP2.receive(Medium_Connection_Request()); { /* uso de una plantilla */ 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>			
Comentarios detallados:			

Se corresponde con:

```
testcase MyTestcase
runs on MyComponentType
{
    var integer MyLocalVar:= 0;
    timer T1 := 15 min;

    default.activate { [expand] OtherwiseFail(); }; /* Activación por
defecto */

    ISAP1.send( ICONreq {} ); /* Definición de plantilla Inline */
    alt {
    [] MSAP2.receive( Medium_Connection_Request() ); { /* uso de una
plantilla */
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_synchl() );
ISAP1.send ( IDISreq {} );
}
[] ISAP1.receive( IDISind {} ); {
verdict.set(inconclusive);
stop();
}
}
};
}
[] MSAP2.receive( MDATindConnection_Request()); {
verdict.set(inconclusive);
stop();
}
[] ISAP1.receive( IDISind {} ); {
verdict.set(inconclusive);
stop();
}
}
};
}
[] ISAP1.receive( IDISind {} ); {
verdict.set(inconclusive);
stop();
}
}
}
}
with display "testcase { purpose " ";};
}
```

8 Tabular presentationformat BNF (BNF del formato de presentación tabular)

```
TabularPresentationFormat ::= ReferenceProforma |
ParametersProforma |
ControlProforma |
ImportsProforma |
EncodingProforma |
SimpleTypesProforma |
StructuredTypesProforma |
PortTypeProforma |
ComponentTypeProforma |
ConstantsProforma |
SignatureProforma |
SimpleTemplatesProforma |
```

```

StructuredTemplatesProforma |
FunctionProforma |
DefaultsProforma |
TeststepProforma |
TestcaseProforma

```

8.1 ReferenceProforma (Proforma de referencia)

```

ReferenceProforma ::= ReferenceKeyword
                    BeginChar
                    ReferenceFieldList
                    EndChar

ReferenceFieldList ::= VersionKeyword AssignmentChar VersionIdentifier SemiColon
                    DateKeyword      AssignmentChar FreeText           SemiColon
                    BaseKeyword      AssignmentChar FreeText           SemiColon
                    TestKeyword      AssignmentChar FreeText           SemiColon
                    PicsKeyword      AssignmentChar FreeText           SemiColon
                    PixitKeyword     AssignmentChar FreeText           SemiColon
                    MethodKeyword    AssignmentChar FreeText           SemiColon
                    [DetailedComments]

VersionIdentifier ::= Number { "." Number }

VersionKeyword ::=      version
DateKeyword    ::=      date
BaseKeyword    ::=      basestandard
TestKeyword    ::=      teststandard
PicsKeyword    ::=      pics
PixitKeyword   ::=      pixit
MethodKeyword  ::=      testmethod
DCommentsKeyword ::=    detailedcomments

DetailedComments ::= DCommentsKeyword      AssignmentChar FreeText

```

8.2 ParametersProforma (Proforma de parámetros)

```

ParametersProforma ::= ParametersKeyword BeginChar
                    ParametersFieldList
                    EndChar

ParameterKeyword ::= parameters

ParameterFieldList ::= [GroupDef]
                    PICSRefList
                    [ParameterCommentsList]
                    [DetailedComments]

GroupDef ::= GroupKeyword      AssignmentChar GroupReference SemiColon
GroupKeyword ::= group
CommentListKeyword ::= commentlist

PICSRefList ::= PicsKeyword PixitKeyword AssignmentChar BeginChar
               [PicsRef {"," PicsRef}]
               EndChar [SemiColon]

PicsRef ::= ModuleParIdentifier AssignmentChar FreeText

ParameterCommentsList ::= CommentListKeyword      AssignmentChar BeginChar
                        [ParComment {"," ParComment}]
                        EndChar [SemiColon]

ParComment ::= ModuleParIdentifier AssignmentChar FreeText

```

8.3 ControlProforma (Proforma de control)

ControlProforma ::= ControlKeyword BeginChar ControlFieldList EndChar

ControlFieldList ::= [VarConstOrTimerCommentList]
[ExtraComments]
[DetailedComments]

8.4 ImportsProforma (Proforma de importaciones)

ImportsProforma ::= ImportsKeyword BeginChar ImportsFieldList EndChar

ImportsKeyword ::= **imports**

ImportType ::= AllKeyword [DefKeyword] | DefKeyword

ImportIdentifier ::= [TypeDefIdentifier | TemplateIdentifier | ConstIdentifier |
TestcaseIdentifier | FunctionIdentifier |
NamedAltIdentifier]

Mark ::= ["*"]

ImportsFieldList ::= [GroupDef]
SourceKeyword AssignmentChar FreeText SemiColon
[SingleComment]
[ExtraCommentList]
[DetailedComments]

SourceKeyword ::= source

SingleComment ::= CommentsKeyword AssignmentChar FreeText SemiColon
CommentsKeyword ::= **comments**

8.5 EncodingProforma (Proforma de codificación)

EncodingProforma ::= EncodingKeyword BeginChar EncodingFieldList EndChar

EncodingKeyword ::= **encoding**

EncodingRuleIdentifier ::= identifier

EncodingFieldList ::= [GroupDef]
EncodingRefList
EncodingDefaultList
[EncodingCommentList]
[DetailedComments]

RefKeyword ::= **reference**

DefaultKeyword ::= **default**

EncodingRefList ::= RefKeyword AssignmentChar BeginChar
[EncodingRef {"", " EncodingRef}]
EndChar [SemiColon]

EncodingRef ::= EncodingRuleIdentifier AssignmentChar FreeText

EncodingDefaultList ::= DefaultKeyword AssignmentChar BeginChar
[EncodingDefault {"", " EncodingDefault}]
EndChar [SemiColon]

EncodingDefault ::= EncodingRuleIdentifier AssignmentChar BooleanExpression

```
EncodingCommentList ::= CommentListKeyword AssignmentChar BeginChar  
                        [EncodingComment {"", " EncodingComment"}]  
                        EndChar [SemiColon]
```

```
EncodingComment ::= EncodingRuleIdentifier AssignmentChar FreeText
```

8.6 SimpleTypesProforma (Proforma de tipos simples)

```
SimpleTypesProforma ::= SimpleTypeKeyword BeginChar SimpleTypeFieldList EndChar  
SimpleTypesKeyword ::= simpletypes
```

```
SimpleTypeFieldList ::= [GroupDef]  
                        [SimpleTypeCommentList]  
                        [DetailedComments]
```

```
SimpleTypeCommentList ::= CommentListKeyword AssignmentChar BeginChar  
                        [SimpleTypeComment {"", " SimpleTypeComment"}]  
                        EndChar [SemiColon]
```

```
SimpleTypeComment ::= SubTypeIdentifier AssignmentChar FreeText
```

8.7 StructuredTypesProforma (Proforma de tipos estructurados)

```
StructuredTypeProforma ::= StructTypeKeyword BeginChar StructTypeFieldList  
                        EndChar
```

```
StructTypeKeyword ::= structuredtype
```

```
StructTypeFieldList ::= [GroupDef]  
                        [SingleComment]  
                        [StructTypeCommentList]  
                        [DetailedComments]
```

```
StructTypeCommentList ::= CommentListKeyword AssignmentChar BeginChar  
                        [StructTypeComment {"", " StructTypeComment"}]  
                        EndChar [SemiColon]
```

```
StructTypeComment ::= StructFieldIdentifier AssignmentChar FreeText
```

```
StructureType ::= record | set | union
```

8.8 PortTypeProforma (Proforma de tipo de puerto)

```
PortTypeProforma ::= PortTypeKeyword BeginChar PortTypeFieldList EndChar  
PortTypeKeyword ::= porttype
```

```
PortTypeFieldList ::= [GroupDef]  
                        [SingleComment]  
                        [PortTypeCommentList]  
                        [DetailedComments]
```

```
PortTypeCommentList ::= CommentListKeyword AssignmentChar BeginChar  
                        [PortTypeComment {"", " PortTypeComment"}]  
                        EndChar [SemiColon]
```

```
PortTypeComment ::= PortTypeIdentifier AssignmentChar FreeText
```

```
PortModelType ::= MessageKeyword | ProcedureKeyword | MixedKeyword  
PortTypeDef ::= BeginChar MixedList {SemiColon MixedList} [SemiColon] EndChar
```


8.9 ComponentTypeProforma (Proforma de tipo de componente)

ComponentTypeProforma ::= ComponentTypeKeyword BeginChar ComponentTypeFieldList
EndChar

ComponentTypeKeyword ::= **componenttype**

ComponentTypeFieldList ::= [GroupDef]
[SingleComment]
[VarConstOrTimerCommentList]
[ExtraComments]
[DetailedComments]

ExtraComments ::= ECommentsKeyword AssignmentChar FreeText SemiColon

ECommentsKeyword ::= **extracomments**

PortList ::= {PortInstance}

TypeOrTimer ::= Type | TimerKeyword

8.10 ConstantsProforma (Proforma de constantes)

ConstantsProforma ::= ConstantsKeyword BeginChar ConstantsFieldList EndChar
ConstantsKeyword ::= **constants**

ConstantsFieldList ::= [GroupDef]
[ConstantsCommentList]
[DetailedComments]

ConstantsCommentList ::= CommentListKeyword AssignmentChar BeginChar
[ConstantsComment {" , " ConstantsComment}]
EndChar [SemiColon]

ConstantsComment ::= ConstIdentifier AssignmentChar FreeText

8.11 SignatureProforma (Proforma de firma)

SignatureProforma ::= SignatureKeyword BeginChar SignatureFieldList EndChar

SignatureFieldList ::= [GroupDef]
[SingleComment]
[ExtraComments]
[DetailedComments]

SignatureIdentifer&ParList ::= SignatureIdentifier "("
[SignatureFormalParList] ")"

8.12 SimpleTemplatesProforma (Proforma de plantillas simples)

SimpleTemplatesProforma ::= SimpleTemplatesKeyword
BeginCharSimpleTemplatesFieldList EndChar
SimpleTemplatesKeyword ::= **simpleTemplates**

SimpleTypeFieldList ::= [GroupDef]
[SimpleTemplatesCommentList]
[DetailedComments]

SimpleTemplatesCommentList ::= CommentListKeyword AssignmentChar BeginChar
[SimpleTemplateComment {" , "
SimpleTemplateComment}]
EndChar [SemiColon]

SimpleTemplateComment ::= TemplateIdentifier AssignmentChar FreeText

SimpleType ::= Type | DerivedDef

/* STATIC SEMANTICS - The referenced type or base template shall not be of a constructed type */

8.13 StructuredTemplatesProforma (Proforma de plantillas estructuradas)

StructuredTemplateProforma ::= StructuredTemplateKeyword BeginChar

TemplateFieldList EndChar

StructuredTemplateKeyword ::= **structuredtemplate**

TemplateFieldList ::= [GroupDef]
[SingleComment]
[TemplateCommentList]
[DetailedComments]

TemplateCommentList ::= CommentListKeyword AssignmentChar BeginChar
[TemplateComment {" , " TemplateComment}]
EndChar [SemiColon]

TemplateComment ::= FieldReference AssignmentChar FreeText

TemplateIdentifier&ParList ::= TemplateIdentifier ["(" TemplateFormalParList)"]

TemplateStructIdentifier ::= Type | Signature | DerivedDef

8.14 FunctionProforma (Proforma de función)

FunctionProforma ::= FunctionKeyword BeginChar FunctionFieldList EndChar

FunctionFieldList ::= [GroupDef]
[SingleComment]
[VarConstOrTimerCommentList]
[ExtraComments]
[DetailedComments]

FunctionIdentifier&ParList ::= FunctionIdentifier ["(" [FunctionFormalParList] ")"]

FunctionCommentList ::= FunctionComment {" , " FunctionComment}

FunctionComment ::= VarConstOrTimerRef AssignmentChar FreeText

VarConstOrTimerCommentList ::= CommentListKeyword AssignmentChar BeginChar
[FunctionCommentList]
EndChar [SemiColon]

VarConstOrTimerIdentifier ::= ConstKeyword ConstIdentifier |
VarIdentifier [ArraySpec] |
TimerIdentifier [ArraySpec]

VarConstOrTimerRef ::= ConstIdentifier | VarIdentifier |
TimerIdentifier

TabularBehaviour ::= FunctionBody

/* STATIC SEMANTICS - The FunctionBody production shall not contain any variable, timer or constant definitions */

8.15 DefaultsProforma (Proforma de valores por defecto)

```
DefaultProforma ::= DefaultKeyword BeginChar DefaultFieldList EndChar
DefaultKeyword ::= default
```

```
DefaultFieldList ::=      [GroupDef]
                        PurposeDef
                        [SingleComment]
                        [DetailedComments]
```

```
PurposeDef ::= PurposeKeyword AssignmentChar FreeText SemiColon
```

```
PurposeKeyword ::= purpose
```

```
NamedAltIdentifier&ParList ::= NamedAltIdentifier ["(" FunctionFormalParList")"]
```

8.16 NamedAltProforma (Proforma de denominación alternativa)

```
NamedAltProforma ::= NamedAltKeyword BeginChar DefaultFieldList EndChar
```

```
NamedAltKeyword ::= namedalt
```

8.17 TestcaseProforma (Proforma de caso de prueba)

```
TestcaseProforma ::= TestcaseKeyword BeginChar TestcaseFieldList EndChar
```

```
TestcaseFieldList ::=      [GroupDef]
                        PurposeDef
                        [SingleComment]
                        [VarOrTimerCommentList]
                        [ExtraComments]
                        [DetailedComments]
```

```
TestcaseIdentifier&ParList ::= TestcaseIdentifier "("
[TestcaseRestrictedFormalParList]"
```

```
TestcaseRestrictedFormalParList ::= FormalVarValuePar {" ," FormalVarValuePar}
```


SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	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	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
Serie X	Redes de datos y comunicación entre sistemas abiertos
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

20816