



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

T.107

(08/95)

TERMINALES PARA SERVICIOS DE TELEMÁTICA

**SERVICIO DE INTERFAZ HOMBRE-MÁQUINA
MEJORADA PARA VIDEOTEX Y OTROS
SERVICIOS DE RECUPERACIÓN (VEMMI)**

Recomendación UIT-T T.107

(Anteriormente «Recomendación del CCITT»)

PREFACIO

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

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

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.º 1 de la CMNT (Helsinki, 1 al 12 de marzo de 1993).

La Recomendación UIT-T T.107 ha sido preparada por la Comisión de Estudio 8 (1993-1996) del UIT-T y fue aprobada por el procedimiento de la Resolución N.º 1 de la CMNT el 11 de agosto de 1995.

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.

© UIT 1996

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

		<i>Página</i>
1	Alcance.....	1
2	Referencias.....	1
3	Definiciones y abreviaturas.....	2
	3.1 Definiciones.....	2
	3.2 Abreviaturas.....	3
4	Modelo general.....	4
	4.1 Introducción.....	4
	4.2 Definición de los elementos VEMMI.....	4
	4.2.1 Definición e identificación de objeto VEMMI.....	5
	4.2.2 Definición de componente VEMMI.....	5
	4.2.3 Definición de ítem de componente VEMMI.....	6
	4.2.4 Definición de recurso.....	6
	4.2.5 Transferencia de objetos VEMMI.....	6
	4.2.6 Transferencia de recursos.....	6
	4.3 Modelo de estructura del plano VEMMI.....	7
	4.3.1 Plano estándar para datos videotex.....	7
	4.3.2 Plano de objetos VEMMI.....	7
	4.4 Modos de operación de los terminales VEMMI.....	7
	4.4.1 Modo estándar.....	7
	4.4.2 Modo VEMMI.....	7
	4.4.3 Conmutación entre el modo videotex estándar y el modo VEMMI.....	9
	4.5 Contenido de datos de elementos VEMMI.....	9
	4.5.1 Definición de datos de texto.....	9
	4.5.2 Mapas de bits.....	10
	4.5.3 Color.....	11
	4.5.4 Datos videotex.....	11
	4.5.5 Sonido.....	11
	4.5.6 Datos gráficos.....	11
	4.5.7 Datos vídeo.....	11
	4.6 Posicionamiento y dimensionamiento de objetos VEMMI.....	11
	4.6.1 Posicionamiento.....	11
	4.6.2 Dimensionamiento.....	12
	4.7 Estados de elementos VEMMI y parámetros de estado.....	13
	4.7.1 Objeto.....	13
	4.7.2 Componente.....	16
	4.8 Gestión de acciones locales.....	17
	4.9 Consideraciones sobre memoria.....	18
	4.10 Reglas comunes para el tratamiento de objetos.....	18
	4.10.1 Estado activo y gestión de enfoque.....	18
	4.10.2 Comportamiento del modo modal.....	19
	4.10.3 Consideraciones relativas al tamaño y recorte.....	19
	4.11 Almacenamiento de objetos locales.....	19
	4.12 Nombres de directorios simbólicos.....	20
	4.13 Reglas específicas para terminales dedicados.....	20
5	Descripción de servicio.....	21
	5.1 Elementos de servicio iniciados por la aplicación VEMMI y el terminal.....	23
	5.2 Elementos de servicio iniciados por la aplicación VEMMI.....	23
	5.2.1 VEMMI_Abrir.....	23
	5.2.2 VEMMI_Cerrar.....	24
	5.2.3 VEMMI_Reanudar.....	24

5.2.4	VEMMI_Suspender	24
5.2.5	VEMMI_Identificar_cap_term	25
5.2.6	VEMMI_Fijar_opciones	25
5.2.7	VEMMI_Crear_objeto	26
5.2.8	VEMMI_Abrir_objeto	26
5.2.9	VEMMI_Abrir_objeto_bloqueante.....	26
5.2.10	VEMMI_Cerrar_objeto.....	27
5.2.11	VEMMI_Cerrar_todo	27
5.2.12	VEMMI_Destruir_objeto.....	27
5.2.13	VEMMI_Inhabilitar_acceso_objeto.....	27
5.2.14	VEMMI_Habilitar_acceso_objeto	28
5.2.15	VEMMI_Modificar_componente	28
5.2.16	VEMMI_Cambiar_ubicación_objeto.....	29
5.2.17	VEMMI_Cargar_tabla_col	29
5.2.18	VEMMI_Restablecer_tabla_col.....	30
5.2.19	VEMMI_Abrir_aplicación.....	31
5.2.20	VEMMI_Borrar_objeto_antiguado.....	31
5.2.21	VEMMI_Almacenar_objetos.....	32
5.2.22	VEMMI_Borrar_objetos.....	32
5.2.23	VEMMI_Bloquear_usuario	32
5.2.24	VEMMI_Desbloquear_usuario.....	32
5.2.25	VEMMI_Transferencia_recursos.....	33
5.3	Elementos de servicio iniciados por el terminal	34
5.3.1	VEMMI_Identificar_resp_cap_term.....	34
5.3.2	VEMMI_Retransmisión_objeto.....	34
5.3.3	VEMMI_Datos_usuario.....	35
5.3.4	VEMMI_Resp_abrir_aplicación	36
5.3.5	VEMMI_Resp_almacenar_objetos.....	36
5.3.6	VEMMI_Error	36
5.3.7	VEMMI_Aborto_transferencia_recursos.....	37
6	Presentación de objeto VEMMI.....	37
6.1	Barra de aplicaciones	38
6.1.1	Composición	38
6.2	Barra de botones	38
6.2.1	Composición	38
6.3	Menú superpuesto.....	38
6.3.1	Composición	38
6.4	Casilla de diálogo	39
6.4.1	Composición	39
6.5	Objeto operativo	40
6.6	Objeto recurso mapa de bits.....	41
6.7	Objeto recurso videotex	41
6.8	Objeto recurso texto.....	41
6.9	Objeto recurso tipo de caracteres.....	41
6.10	Objeto metacódigo.....	41
6.11	Casilla de mensajes.....	41
7	Descripción funcional	41
7.1	Reglas generales para el comportamiento de los elementos	41
7.1.1	Interacción de usuario	41
7.1.2	Acciones locales e informes.....	41
7.1.3	Relación entre objetos y componentes.....	42
7.1.4	Abrir/cerrar objetos audio, vídeo, recurso y metacódigo	43
7.1.5	Operación maximizar.....	43
7.1.6	Convenios de notación.....	44
7.1.7	Mnemónica	44

	<i>Página</i>
7.2	Formato de texto 44
7.2.1	Texto VEMMI de alta calidad 44
7.2.2	Etiquetas y títulos de texto 45
7.3	Barra de aplicaciones 46
7.3.1	Composición 48
7.4	Barra de botones 53
7.4.1	Composición 54
7.5	Menú superpuesto 55
7.5.1	Composición 57
7.6	Casilla de diálogo 58
7.6.1	Composición 61
7.7	Casilla de mensaje 83
7.8	Objeto operativo 85
7.9	Objeto recurso mapa de bits 87
7.10	Objeto de recurso videotex 88
7.11	Objeto recurso texto 88
7.12	Objeto recurso tipo de caracteres 89
7.13	Objeto metacódigo 89
7.14	Definición de tipo de datos de mapa de bits VEMMI 89
7.15	Catálogo de identificación de codificación de contenidos VEMMI 89
8	Representación codificada completa de VEMMI 91
8.1	Introducción 91
8.2	Notación utilizada 91
8.3	Conmutación global del entorno de codificación 91
8.3.1	Conmutación al modo VEMMI 91
8.3.2	Estructura de la sintaxis ISO/CEI 9281 [14] 93
8.4	Sintaxis de las instrucciones VEMMI 95
8.5	Objetos y componentes 98
8.6	Acciones locales 102
9	Codificación 103
9.1	Estructura de las instrucciones 103
9.2	Estructura de objeto, componente y atributo 103
9.3	Codificación de símbolos de terminal 104
9.3.1	Códigos operacionales 104
9.3.2	Enteros 104
9.3.3	Enumerado 105
9.3.4	Cadenas 105
9.3.5	NDC 106
9.4	Atributos y símbolos de nivel inferior 107
9.5	Códigos operacionales (opcódigos) 108
9.6	Sintaxis de VEMMI_Modificar_componente 111
9.7	Valores por defecto 113
10	Introducción del servicio VEMMI en las Recomendaciones existentes relativas a videotex 116
10.1	Introducción de VEMMI en la Recomendación T.101 [4] 116
10.2	Introducción de VEMMI en la Recomendación T.105 [6] 116
Anexo A	– Cadena T.51 116
A.1	Alcance 116
A.2	Juegos de caracteres gráficos 116
A.3	Técnica de extensión de código 119
A.4	Repertorio del juego de caracteres latinos 119
A.5	Funciones de control 119
Anexo B	– Subconjunto obligatorio de ISO 8859 [13] 120
Anexo C	– Núcleo mínimo de tipos de datos 121

SUMARIO

La presente Recomendación especifica la sintaxis que han de utilizar los servicios videotex u otros servicios de recuperación para la implementación de una interfaz hombre-máquina mejorada (VEMMI, *videotex enhanced man machine interface*).

La VEMMI es un medio de mejorar la ergonomía e interactividad de los sistemas de recuperación que utilizan elementos de diálogo gráficos tales como barra de aplicaciones, menús superpuestos, casilla de diálogo.

El plano estándar continúa recibiendo los datos estándar de la plataforma subyacente. Por consiguiente, en un terminal VEMMI pueden ejecutarse también aplicaciones videotex estándar. El plano de objetos VEMMI recibe objetos VEMMI. Se proporciona un mecanismo de conmutación automática entre el modo estándar y el modo VEMMI.

Esta Recomendación define tipos de datos (texto de alta calidad VEMMI, mapas de bits VEMMI) y permite la integración de otros tipos de datos ampliamente utilizados (por ejemplo, JPEG, BMP, WAVE, MIDI, videotex) en el diálogo VEMMI. Se definen también objetos operativos para extender las capacidades de una aplicación VEMMI durante el tiempo de ejecución.

Los objetos VEMMI pueden distribuirse entre un computador principal y un terminal VEMMI para mejorar las prestaciones de las aplicaciones interactivas. La aplicación VEMMI puede controlar (salvar, cargar, actualizar) objetos VEMMI en el terminal.

La presente Recomendación contiene la descripción de servicio, los elementos de servicio y su codificación.

SERVICIO DE INTERFAZ HOMBRE-MÁQUINA MEJORADA PARA VIDEOTEX Y OTROS SERVICIOS DE RECUPERACIÓN (VEMMI)

(Ginebra, 1995)

1 Alcance

La presente Recomendación especifica la sintaxis de datos que han de utilizar los servicios videotex y los servicios de recuperación de información multimedia/hipermedia para la implementación de la interfaz hombre-máquina mejorada para videotex (VEMMI).

En el caso del videotex, la presente Recomendación es aplicable al servicio videotex y a los terminales videotex asociados. Estos terminales pueden estar conectados al servicio videotex por la red telefónica pública conmutada (RTPC), la red digital de servicios integrados (RDSI) o la red pública de datos con conmutación de paquetes (RPDCP). Los terminales deberán soportar normalmente el videotex basado en sintaxis.

La presente Recomendación puede también utilizarse con cualquier clase de servicio de recuperación (no relacionado con el videotex) utilizando la correspondiente plataforma subyacente y los correspondientes tipos de datos de contenido.

2 Referencias

Las siguientes Recomendaciones UIT-T y otras referencias contienen disposiciones que, mediante la referencia hecha en este texto, constituyen disposiciones de la presente Recomendación. En el momento de la publicación, las ediciones indicadas eran válidas. Todas las Recomendaciones y otras referencias son objeto de revisiones, por lo que se preconiza que todos los usuarios de esta Recomendación investiguen la posibilidad de aplicar la edición más reciente de las Recomendaciones y otras referencias enumeradas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T vigentes.

- [1] Recomendación T.50 del CCITT (1992), *Alfabeto internacional de referencia (anteriormente alfabeto internacional N.º 5 o IA5) – Tecnología de la información – Juego de caracteres codificado de siete bits para intercambio de información.*
- [2] Recomendación T.51 del CCITT (1992), *Juegos de caracteres codificados basados en el alfabeto latino para los servicios de telemática.*
- [3] Recomendación UIT-T T.52 (1993), *Juegos de caracteres codificados no latinos para los servicios de telemática.*
- [4] Recomendación UIT-T T.101 (1994), *Interfuncionamiento internacional de servicios videotex.*
- [5] Recomendación UIT-T T.102 (1993), *Protocolos de extremo a extremo de videotex basado en sintaxis en modo circuito en la red digital de servicios integrados.*
- [6] Recomendación UIT-T T.105 (1994), *Protocolo de la capa de aplicación para videotex basado en sintaxis.*
- [7] Recomendación UIT-T H.261 (1993), *Códec vídeo para servicios audiovisuales a $p \times 64$ kbit/s.*
- [8] Recomendación UIT-T H.320 (1993), *Sistemas y equipos terminales videotelefónicos de banda estrecha.*
- [9] Recomendación UIT-T F.300 (1993), *Servicio videotex.*
- [10] ISO 2022:1986, *Information processing – ISO 7-bit and 8-bit coded character sets – Code extension techniques.*
- [11] ISO 2375:1985, *Data processing – Procedure for registration of escape sequences.*
- [12] ISO 8632:1992, *Information technology – Computer graphics – Metafile for storage and transfer of picture description information.*
- [13] ISO 8859:1987, *Information processing – 8-bit single-byte coded graphic character sets.*
- [14] ISO 9281:1990, *Information technology – Picture coding methods.*
- [15] ISO 10646:1993, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and basic multilingual plane.*

- [16] ISO-IS 10918-1:1994, *Digital compression and coding of continuous-tone still images. Part 1: Requirements and guidelines.*
- [17] ISO 11172-1:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media up to about 1,5 Mbit/s; Part 1: Systems.*
- [18] ISO 11172-2:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media up to about 1,5 Mbit/s; Part 2: Video.*

3 Definiciones y abreviaturas

3.1 Definiciones

A los efectos de la presente Recomendación, se aplican las siguientes definiciones:

- 3.1.1 controles:** Elementos visuales de la interfaz de usuario que permiten a un usuario interactuar con datos.
- 3.1.2 zona de visualización definida:** Véase la Recomendación F.300 [9].
- 3.1.3 realce:** Modo de resaltar, cambio de color u otra indicación visible de la condición de un elemento u opción y el efecto de esa condición sobre la capacidad de un usuario de interactuar con ese elemento. El realce puede también dar información adicional sobre el estado de un objeto. El método utilizado para realzar un elemento depende del terminal.
- 3.1.4 etiqueta:** Datos de texto asociados con un componente VEMMI, para comunicar al usuario la finalidad de un componente o ítem determinado.
- 3.1.5 gestor local:** Véase gestor local VEMMI.
- 3.1.6 mnemónica:** Carácter alfanumérico fácil de recordar que activa un componente de selección por menú VEMMI y lo valida. Un carácter mnemónico puede también utilizarse para validar un botón pulsado activo.
- 3.1.7 modo modal:** Cuando un objeto VEMMI es «modal», al usuario no le es posible dejar este objeto para pasar a otro objeto VEMMI de la misma aplicación con las diferentes herramientas de acceso posibles. Todo intento de acceder a otro objeto es rechazado y posiblemente indicado al usuario mediante una señal sonora.
- 3.1.8 transferencia de recursos:** Mecanismo para transferir ficheros referenciados por objetos de recurso VEMMI desde una aplicación VEMMI a un terminal VEMMI.
- 3.1.9 presentación elástica:** Visualización reducida o ampliada de un mapa de bits para cumplir determinados requisitos de espacio.
- 3.1.10 presentación enlosada:** Visualización repetida de un determinado mapa de bits en dirección horizontal y/o vertical a fin de cumplir determinados requisitos de espacio.
- 3.1.11 aplicación videotex:** Aplicación videotex que utiliza datos codificados, protocolos y perfiles, definidos en las Recomendaciones relativas a videotex referenciadas en la cláusula 2. Una aplicación videotex estándar no utiliza un servicio datos y protocolos VEMMI (véase la Recomendación F.300 [9]).
- 3.1.12 datos videotex:** Datos intercambiados entre una aplicación videotex y un terminal videotex.
- 3.1.13 validación:** Acción de activación por el usuario seguida por una confirmación de la opción con un teclado o con un dispositivo de apuntamiento.
- 3.1.14 aplicación VEMMI:** Aplicación que ofrece una interfaz hombre-máquina mejorada, que se describe en la presente Recomendación.
- 3.1.15 datos VEMMI:** Descripción y contenido de objetos VEMMI e instrucciones VEMMI intercambiadas entre la aplicación VEMMI y el terminal VEMMI.
- 3.1.16 gestor local VEMMI:** Soporte lógico que funciona en el terminal VEMMI para tratar y presentar los objetos VEMMI que son enviados al usuario por la aplicación VEMMI.
- 3.1.17 terminal VEMMI:** Terminal videotex que puede ejecutar un gestor local VEMMI.
- 3.1.18 computador principal videotex:** Véase la Recomendación F.300 [9].
- 3.1.19 terminal videotex:** Véase la Recomendación F.300 [9].

3.2 Abreviaturas

A los efectos de la presente Recomendación, se aplican las siguientes abreviaturas:

BIN	Número de identificación de mapa de bits (<i>bitmap identification number</i>)
BMP	Mapa de bits independiente del dispositivo Microsoft Windows (<i>microsoft windows device-independent bitmap</i>)
CCITT	Comité Consultivo Internacional Telegráfico y Telefónico
CD-ROM	Disco compacto-Memoria de lectura solamente (<i>compact disk-read only memory</i>)
CEI	Comisión Electrotécnica Internacional
CGM	Metaarchivo de gráficos por computador (<i>computer graphics metafile</i>)
CIN	Número de identificación de componente (<i>component identification number</i>)
CMI	Identificador de método de codificación (<i>coding method identifier</i>)
CR	Retorno del carro (<i>carriage return</i>)
DDA	Zona de visualización definida (<i>defined display area</i>)
DIB	Mapa de bits independiente del dispositivo (<i>device-independent bitmap</i>)
DRCS	Conjunto de caracteres dinámicamente redefinibles (<i>dynamically redefinable character set</i>)
DS I	Sintaxis de datos conforme al Anexo B/T.101 [4]
DS II	Sintaxis de datos conforme al Anexo C/T.101 [4]
DS III	Sintaxis de datos conforme al Anexo D/T.101 [4]
ESC	Escape (<i>escape</i>)
ETS	Norma de telecomunicaciones europea (<i>european telecommunication standard</i>)
ETSI	Instituto Europeo de Normalización de las Telecomunicaciones (<i>European Telecommunications Standards Institute</i>)
FIN	Número de identificación de tipo de caracteres (<i>font identification number</i>)
G0	Juego de caracteres primario de la Recomendación T.51 [2]
G2	Juego de caracteres suplementario de la Recomendación T.51 [2]
GIF	Formato de intercambio de gráficos (<i>graphics interchange format</i>)
GMT	Hora media de Greenwich (<i>Greenwich mean time</i>)
GUI	Interfaz de usuario gráfico (<i>graphical user interface</i>)
IRV	Versión internacional de referencia (<i>international reference version</i>)
IS	Norma internacional (<i>international standard</i>)
ISO	Organización Internacional de Normalización (<i>International Organization for Standardization</i>)
JIS	Instituto Japonés de Normalización (<i>Japanese institute for standardization</i>)
JPEG	Grupo Mixto de Expertos sobre fotografía (<i>joint photographic experts group</i>)
LF	Cambio de renglón (<i>line feed</i>)
LI	Indicador de longitud (<i>length indicator</i>)
MDI	Indicador más datos (<i>more data indicator</i>)
MIDI	Interfaz digital para instrumentos musicales (<i>musical instrument digital interface</i>)
MPEG	Grupo de expertos sobre imágenes en movimiento (<i>moving picture experts group</i>)
NDC	Coordenada de dispositivo normalizada (<i>normalized device coordinate</i>)

OIN	Número de identificación de objeto (<i>object identification number</i>)
PCD	Delimitador de código de imagen (<i>picture code delimiter</i>)
PCE	Entidad de control de imagen (<i>picture control entity</i>)
PDE	Entidad de datos de imagen (<i>picture data entity</i>)
PE	Elemento de imagen (<i>picture element</i>)
PI	Identificador de imagen (<i>picture identifier</i>)
PM	Modo de imagen (<i>picture mode</i>)
RDSI	Red digital de servicios integrados
RGB	Rojo, verde, azul (<i>red, green, blue</i>)
RPDCP	Red pública de datos con conmutación de paquetes
RTPC	Red telefónica pública conmutada
SBV	Videotex basado en sintaxis (<i>syntax-based videotex</i>)
TE	Equipo terminal (<i>terminal equipment</i>)
TFI	Identificador de facilidad de terminal (<i>terminal facility identifier</i>)
TIN	Número de identificación de texto (<i>text identification number</i>)
TLV	Valor de longitud de tipo (<i>type length value</i>)
TV	Televisión
UI	Interfaz de usuario (<i>user interface</i>)
UIT-T	Unión Internacional de Telecomunicaciones – Sector de Normalización
VEMMI	Interfaz hombre-máquina mejorada videotex (<i>videotex enhanced man machine interface</i>)
VIN	Número de identificación videotex (<i>videotex identification number</i>)
VPDE	Elemento de datos de presentación videotex (<i>videotex presentation data element</i>)
VTX	Videotex

4 Modelo general

4.1 Introducción

Entre un computador principal y un terminal VEMMI, el servicio VEMMI trata:

- objetos VEMMI generales, descritos en la presente Recomendación;
- contenido de datos definido en esta Recomendación;
- contenido de datos referenciado en esta Recomendación.

Un terminal VEMMI puede tratar también una aplicación videotex estándar que utiliza datos codificados y protocolos descritos en las Recomendaciones relativas a videotex referenciadas en la cláusula 2.

4.2 Definición de los elementos VEMMI

Las unidades lógicas que forman la estructura de VEMMI se denominarán y definirán como sigue:

- objetos VEMMI u objetos;
- componentes VEMMI o componentes;
- ítem de componente VEMMI o ítems.

El elemento VEMMI es un nombre genérico utilizado en la presente Recomendación para designar un objeto, un componente o un ítem.

En la Figura 1 se muestra un ejemplo:

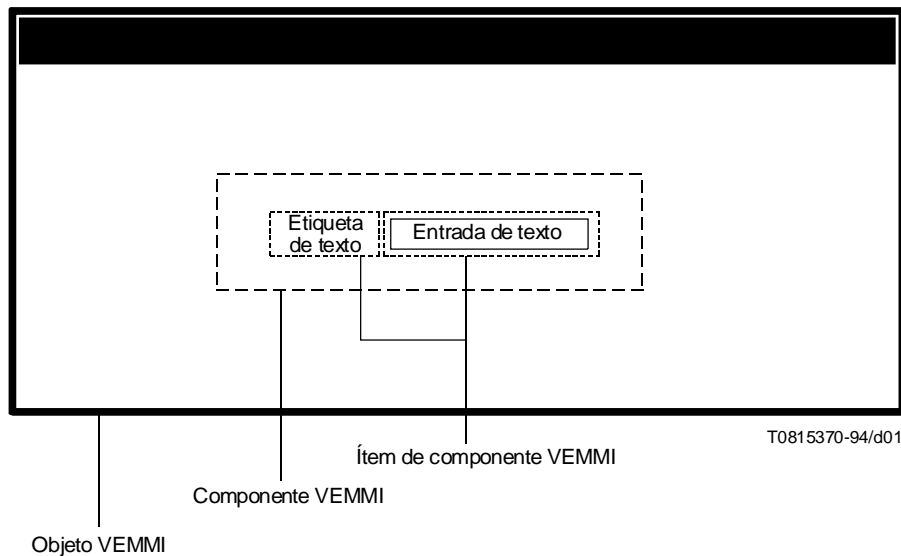


FIGURA 1/T.107

Ejemplo que muestra objetos/componentes/ítems

4.2.1 Definición e identificación de objeto VEMMI

Se definen en esta Recomendación los cuatro diferentes tipos de objetos VEMMI siguientes:

- objetos de visualización;
- objetos operativos;
- objetos de recurso;
- objetos de metacódigo.

Si no se indica otra cosa, el término objeto utilizado independientemente designa también un objeto de visualización.

Los objetos VEMMI son las unidades lógicas utilizadas por una aplicación VEMMI para interactuar con el usuario.

Los objetos VEMMI están formados por diferentes componentes.

Los objetos se definen solamente considerando su funcionalidad, su tamaño y su posición con respecto a la zona de visualización definida (DDA). La representación de los objetos depende del terminal.

Cada objeto será identificado por un número de identificación de objeto (OIN) que será único dentro de una aplicación VEMMI a la vez.

4.2.2 Definición de componente VEMMI

Los componentes VEMMI pertenecen siempre a un objeto VEMMI y sólo son válidos dentro de este objeto. El objeto al cual pertenece un componente se denomina objeto progenitor.

Para transportar información, los componentes pueden llevar un contenido de datos (véase 4.5).

Los componentes sólo se definen en cuanto a su funcionalidad, tipo de contenido, y tamaño y posición relativos al objeto. La representación de los componentes depende del terminal. La representación del contenido de datos se especifica mediante esta Recomendación (para los tipos de datos que se definen en esta Recomendación), o mediante la norma de presentación correspondiente (para los tipos de datos que se definen en otras Recomendaciones).

Cada componente será identificado por un número de identificación de componente (CIN) que será único dentro de un objeto.

4.2.3 Definición de ítem de componente VEMMI

La subunidad de un componente VEMMI es un ítem de componente. Cada ítem es una parte integrante de un componente. La definición de un ítem de componente sólo es válida dentro de este componente.

4.2.4 Definición de recurso

Los recursos son elementos que pueden ser referenciados por componentes u objetos. Un recurso puede ser referenciado por más de un elemento. Se definen los recursos siguientes:

- la tabla de colores, que es única para cada aplicación;
- los archivos almacenados en un terminal (identificados por el nombre de archivo o fichero) pueden contener datos de sonido, objetos operativos, etc.;
- una combinación de tipo de caracteres y un conjunto de atributos es un objeto de recurso. Se identifica mediante un número de identificación de tipo de caracteres (FIN);
- un mapa de bits es un objeto de recurso. Se identifica mediante un número de identificación de mapa de bits (BIN);
- el texto puede ser un objeto de recurso. Se identifica mediante un número de identificación de texto (TIN);
- el videotex puede ser un objeto de recurso. Se identifica mediante un número de identificación videotex (VIN);
- conjuntos de objetos almacenados en el terminal entre dos sesiones (se identifican mediante atributos).

NOTA – FIN, BIN, VIN son los números de identificación de objeto (OIN). Los términos FIN, BIN y VIN se utilizan solamente para indicar claramente que el objeto correspondiente es un objeto de recurso.

4.2.5 Transferencia de objetos VEMMI

Los objetos VEMMI pueden transmitirse al terminal VEMMI utilizando una red de telecomunicación. Si se almacenan entonces en el terminal, se convierten en objetos locales. Los objetos VEMMI pueden también telecargarse utilizando cualquier transferencia de ficheros. Se convierten en objetos locales también y son tratados del mismo modo que objetos transferidos dentro del diálogo VEMMI y almacenados por la primitiva de servicio correspondiente. Los objetos VEMMI pueden también transferirse al terminal VEMMI por correo postal (CD-ROM, disquetes, etc.).

4.2.6 Transferencia de recursos

Los objetos de recursos VEMMI pueden referenciar ficheros que contienen los datos de visualización de recursos. Estos ficheros se denominan ficheros de datos de recursos llamados.

VEMMI especifica el modo en que los ficheros de datos de recursos son transmitidos al terminal (transferencia de recursos VEMMI). Los ficheros de datos de recursos VEMMI pueden también transferirse al terminal VEMMI utilizando la transferencia de ficheros utilizada en el servicio estándar en el que opera el terminal o por correo postal (CD-ROM, disquetes, etc.).

A fin de ofrecer un nivel de calidad satisfactorio y características adaptadas a la funcionalidad VEMMI, el mecanismo de transferencia de recursos ofrece las facilidades siguientes:

- la transferencia de recursos puede efectuarse sin que el usuario lo sepa;
- la transferencia de recursos puede efectuarse como tarea paralela. Durante la transferencia de recursos, el usuario puede seguir interactuando con la aplicación VEMMI (aunque si la velocidad de la red no es suficiente, el tiempo de respuesta del servidor puede ser afectada negativamente por la operación de transferencia de recursos);
- la aplicación puede abrir un objeto VEMMI (ventana) que visualice la situación actual de la transferencia (por ejemplo, un gráfico que represente el porcentaje del recurso copiado), y ofrecer opcionalmente un método para cancelar, mantener o reanudar la transferencia (por ejemplo, mediante botones);
- pueden realizarse de manera independiente varias transferencias de recursos;
- posible interacción de usuario en la operación de transferencia de recursos (aborto).

El soporte de recursos es opcional para un terminal VEMMI. Si el terminal soporta la facilidad de almacenamiento local, soportará igualmente la transferencia de recursos VEMMI.

4.3 Modelo de estructura del plano VEMMI

El modelo de visualización consta de dos planos independientes:

- la memoria de datos estándar y la ventana de datos estándar;
- la memoria de datos VEMMI y la ventana de datos VEMMI.

Este modelo y una posible estructura del terminal se presentan en la Figura 2.

Un terminal VEMMI implementará el comportamiento de este modelo de visualización. Sin embargo, no se hace ninguna hipótesis sobre la estructura del plano físico real ni sobre cómo implementa el terminal ese modelo de estructura de plano.

El plano estándar es el área de salida donde el servicio de recuperación utilizado como plataforma para el servicio VEMMI visualiza sus datos. Si se utiliza un servicio videotex como plataforma, el plano lógico estándar es la DDA utilizada por una aplicación videotex ordinaria. En las subcláusulas siguientes se supone la utilización de videotex como plataforma. Sin embargo, todas las reglas son aplicables aunque se use una plataforma subyacente diferente de videotex (por ejemplo, VT100, canal de televisión).

Los dos planos de salida son independientes en cuanto a sus posiciones y dimensiones en la pantalla del terminal.

4.3.1 Plano estándar para datos videotex

Este plano lógico recibe datos videotex estándar. Continuará soportando las reglas de datos videotex y prioridades definidas en las Recomendaciones relativas al servicio videotex referenciadas en la cláusula 2.

4.3.2 Plano de objetos VEMMI

Este plano lógico recibe objetos VEMMI descritos en esta Recomendación, y sus contenidos de datos codificados. El plano de objetos VEMMI será capaz de tratar los mecanismos de superposición y restablecimiento de objetos referenciados en 4.6 y 4.10.1. Dentro de este plano, y para un contenido de datos dado, se aplican las reglas y prioridades de datos definidas en las normas correspondientes.

4.4 Modos de operación de los terminales VEMMI

Con independencia de su modo de operación vigente, el terminal proporcionará siempre herramientas específicas para gestionar las funciones específicas de plataforma (desconexión, etc.)

4.4.1 Modo estándar

Éste es el modo de operación inicial de un terminal VEMMI cuando se enciende o reinicia.

En el modo estándar, el terminal VEMMI visualiza datos estándar en el plano estándar. Este tipo de datos estándar depende del servicio de recuperación al que se conecta el terminal VEMMI. Los datos videotex estándar son totalmente visibles; el plano lógico de objetos VEMMI es transparente. Las entradas de usuario no son controladas por el gestor local VEMMI.

El terminal ignorará todas las instrucciones VEMMI, salvo VEMMI_Abrir o VEMMI_Reanudar si la sesión VEMMI se suspendió anteriormente con la VEMMI_Suspender. Esta instrucción ejecutará una conmutación al modo VEMMI (luego, los conmutadores VEMMI On/Off se ponen en la posición ON).

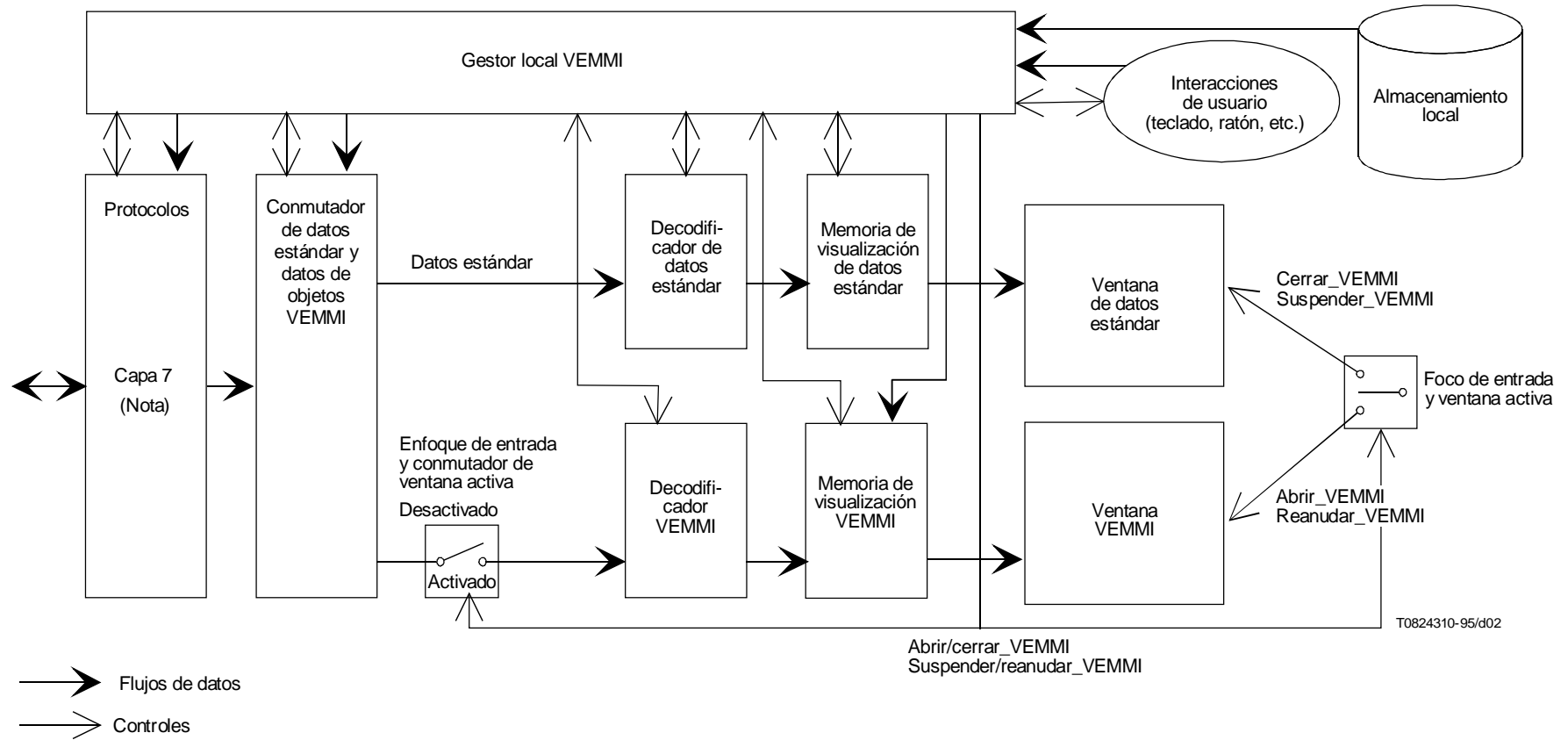
4.4.2 Modo VEMMI

En el modo VEMMI, un terminal VEMMI puede recibir datos videotex estándar y datos de objetos VEMMI en dos trayectos paralelos, como se muestra en la Figura 2.

Los datos de objetos VEMMI son visualizados en el plano de objetos VEMMI. Los datos videotex estándar son visualizados en el plano estándar. El orden de visualización de objetos y de componentes dentro de objetos, corresponderá a su orden de transmisión (último recibido, último visualizado).

En el modo VEMMI, los datos estándar recibidos se utilizan para actualizar la memoria de visualización de datos estándar. Una actualización de la ventana de datos estándar no causa su activación.

Las entradas de usuario son controladas por el gestor local VEMMI en la ventana VEMMI, que es la ventana activa en este modo.



NOTA – Los protocolos videotex estándar incluyen protocolos basados en el SBV o protocolos modificados para el modo VEMMI definido en la cláusula 5.

FIGURA 2/T.107
Ejemplo de posible estructura del terminal VEMMI

4.4.3 Conmutación entre el modo videotex estándar y el modo VEMMI

Desde el modo videotex estándar, un terminal VEMMI se conmutará al modo VEMMI utilizando la instrucción VEMMI_Abrir o VEMMI_Reanudar.

Desde el modo inicial, el terminal puede conmutarse al modo VEMMI utilizando la instrucción VEMMI_Abrir.

Cuando se recibe una instrucción VEMMI_Cerrar, el terminal VEMMI conmutará al modo de operación videotex estándar.

Las instrucciones VEMMI_Reanudar y la instrucción VEMMI_Suspender pueden utilizarse para conmutar temporalmente entre los dos modos.

4.5 Contenido de datos de elementos VEMMI

En esta Recomendación se definen los tipos de datos y los atributos de contenido siguientes:

- a) texto VEMMI de alta calidad;
- b) mapas de bits VEMMI;
- c) color.

Los siguientes tipos de datos y atributos de contenido se definen fuera de esta Recomendación, pero se utilizan como contenido de elementos VEMMI de esta Recomendación:

- a) datos videotex (como contenido de datos opcional de la zona de salida gráfica);
- b) mapa de bits (JPEG, GIF, BMP);
- c) sonido (WAVE, MIDI);
- d) datos gráficos;
- e) datos vídeo.

Es obligatorio que un terminal VEMMI entienda todas las instrucciones relacionadas con el contenido de datos antes mencionado.

No es obligatorio utilizar un determinado tipo de datos de contenido en una aplicación VEMMI. Sin embargo, a fin de proporcionar un servicio VEMMI satisfactorio, un terminal debe como requisito mínimo soportar los tipos de datos indicados en el anexo C.

4.5.1 Definición de datos de texto

Los datos de texto se codifican de acuerdo con la serie ISO 8859 [13]. Según la implementación del terminal, sólo podrá soportarse una parte de la serie 8859 (véase en el anexo B el subconjunto obligatorio 8859 de la serie ISO 8859). Además, la aplicación anfitriona puede utilizar la instrucción VEMMI_Identificar_Cap_Term para preguntar sobre los juegos de caracteres de caracteres soportados en el terminal. Puede seleccionarse un conjunto de caracteres determinado utilizando la instrucción VEMMI_Fijar_Opciones. Esta codificación de datos de texto se considera como la codificación de texto básico. Cuando se utilizan varios juegos de caracteres en el misma aplicación (aplicación multilingüe), la aplicación puede utilizar las secuencias de escape de designación apropiadas ISO registradas, definidas en ISO 2022 [10] y en la serie ISO 8859 [13] para conmutar entre los diferentes juegos de caracteres.

Pueden utilizarse opcionalmente las codificaciones de datos de texto siguientes:

- ISO 10646-1 Unicode [15];
- Recomendación T.51 [2], indicada en el anexo A;
- Recomendación T.52 [3].

Para visualizar títulos de ventana, opciones de menú y etiquetas, la aplicación de terminal puede utilizar el tipo de caracteres (font, en inglés; también llamada «fuente» en español) del sistema local. Para salida de texto específica de la aplicación, VEMMI ofrece capacidades mejoradas. Sin embargo, los terminales que operan con texto simple pueden utilizar el tipo de caracteres del sistema para visualizar el texto de alta calidad. Estos terminales podrán entonces ignorar los atributos de texto que no puedan procesar y visualizar el texto utilizando el tipo de caracteres del sistema, en el color disponible del sistema y sobre el fondo del sistema.

Los datos de texto pueden ser parte de una definición de objeto o de componente, o pueden definirse como un objeto de recurso. Se especifican los atributos siguientes:

- color del texto;
- tipo de caracteres del texto;
- altura del texto.

Color del texto define el color de los caracteres de primer plano. El fondo de los caracteres es específico del terminal. El **tipo de caracteres del texto** es un juego de caracteres con un determinado diseño similar de los caracteres. En este contexto, una aplicación puede elegir una familia de tipos de caracteres, y la aplicación de terminal elegirá el tipo de caracteres específico de entre los que componen esa familia (en esta Recomendación, «tipo de caracteres» significa siempre una «familia» de esos tipos). En el Cuadro 1 se presentan las familias de tipos de caracteres VEMMI:

CUADRO 1/T.107

Familias de tipos de letra VEMMI

Familia de tipo de caracteres	Características del tipo de caracteres	Ejemplo
SWISS	Proporcional, sin serifs	Helvetica, Switzerland
ROMAN	Proporcional, con serifs	Times roman
FIXFONT	Un espacio	Courier

Los tipos de caracteres de estas familias soportarán diversos tamaños (tipos graduables).

La **altura del texto** especifica la altura del tipo de caracteres. Se trata de la altura de la célula del carácter, incluido un posible avance interno. El gestor local insertará un espacio adicional entre filas de texto adyacentes. La altura se mide en puntos (1/72 de pulgada). Un punto corresponde a 1/400 NDC. Los caracteres en cursiva necesitan el atributo **cursiva**. La **negrita** aumenta el grosor de la línea, y el **subrayado** traza una línea debajo de cada carácter.

Ejemplo:

SWISS, 10 puntos y negrita

Roman, 8 puntos y subrayado

Roman, 16 puntos, cursiva

FIXFONT, 12 puntos, blanco

Una combinación de un tipo de caracteres y un conjunto de atributos se denomina también «tipo de caracteres atribuido», y se trata como un objeto de recurso tipo de caracteres. Se referencia mediante el FIN (número de identificación de tipo de caracteres).

Deben utilizarse los siguientes caracteres de control para formatear el texto correctamente:

- SP, NBSP, SHY: véase ISO 8859-1 [13];
- CR, LF: véase la Recomendación T.50 [1].

Para proporcionar la funcionalidad «cambio de renglón», en el contenido de texto de un componente o un ítem de componente, se usarán los caracteres de control «retorno del carro» (CR) + cambio de renglón (LF), de la Recomendación T.50 [1].

4.5.2 Mapas de bits

Un mapa de bits (bitmap), es una matriz de pixels que contiene índices de color, que indican colores de la tabla de colores, o de componentes rojo-verde-azul (RGB). Un mapa de bits puede ser referenciado por un objeto o un componente, y cuando es instanciado por los terminales, aparece como un rectángulo con un patrón de colores de la matriz de pixels correspondiente. La definición es casi siempre independiente del dispositivo, mientras que la relación entre los bits del mapa y los pixels del dispositivo depende del dispositivo.

El mapa de bits se trata como un objeto de recurso y es referenciado por objetos o componentes mediante el «número de identificación de mapa de bits» (BIN), que es único en una aplicación. En un punto de tiempo dado pueden definirse varios mapas de bits.

4.5.3 Color

La tabla de color proporciona un método para acceder a las capacidades de color del dispositivo terminal. Se supone que el dispositivo puede presentar al menos 256 colores simultáneamente. Puesto que el dispositivo tabla de color es compartido por más de una aplicación, las UI suministran mecanismos (por ejemplo, paletas de color lógicas) para soportar cada aplicación con su propia tabla. Esta subcláusula define la tabla de colores de la aplicación VEMMI.

De los 256 colores, la interfaz de usuario (UI) reserva 20 colores para una tabla de sistema. Por consiguiente, el número máximo de colores de la tabla de colores VEMMI es 236. En la práctica, una aplicación utilizará sólo unos pocos colores. La aplicación anfitriona tiene la posibilidad de definir la tabla de colores enviando componentes RGB. Los objetos, componentes o mapas de bits pueden seleccionar los colores mediante índices. Por defecto, 16 entradas de color (0 a 15) están predefinidas. Si la aplicación utiliza un índice de color mayor que 15, que no fue definido, el color visualizado depende del terminal.

Otras aplicaciones activas en el terminal pueden estar utilizando entradas de color. Esto tiene que gestionarlo la UI local.

Los terminales monocromos soportan dos índices de color (por ejemplo, blanco y negro, o la tabla de color cargada con tonos de un color).

4.5.4 Datos videotex

Datos codificados según el Anexo B/T.101, el Anexo C/T.101 y el Anexo D/T.101 [4], por ejemplo, texto, caracteres geométricos.

Datos codificados según el Anexo E/T.101 [4] para datos audio.

Datos codificados según el Anexo E/T.101 [4] para datos fotográficos.

El soporte de datos videotex es opcional para un terminal VEMMI.

La aplicación anfitriona puede preguntar sobre el soporte de estos tipos de datos utilizando la instrucción VEMMI_Identificar_Cap_Term.

4.5.5 Sonido

Una aplicación puede insertar referencias a secuencias de sonido en los objetos. El terminal almacena el sonido en ficheros. Cuando el usuario ejecuta una acción, se procesa el sonido, posiblemente con una tarjeta de interfaz audio.

Los dos formatos de sonido son adecuados para:

- dispositivos de onda (WAVE);
- dispositivos de interfaz digital de instrumentos musicales (MIDI).

El sonido puede también proporcionarse mediante la sintaxis de audio videotex definida como contenido de datos videotex (véase 4.5.4).

4.5.6 Datos gráficos

Los datos gráficos, como líneas, arcos, áreas de relleno y los atributos de dibujo correspondientes no son soportados directamente por esta Recomendación. A pesar de ello, una aplicación VEMMI puede usar programas VEMMI-integrados (por medio de objetos operativos), en los que se llevan a cabo las operaciones de dibujo gráfico.

4.5.7 Datos vídeo

Esta Recomendación no describe los formatos de los datos vídeo, pero proporciona el contenedor (objeto) que puede utilizarse para incorporar imágenes en movimiento en las aplicaciones VEMMI.

4.6 Posicionamiento y dimensionamiento de objetos VEMMI

4.6.1 Posicionamiento

El plano estándar seguirá soportando el sistema de coordenadas utilizado en la plataforma subyacente.

El plano de objetos VEMMI soportará un sistema de coordenadas de dispositivo normalizado (NDC) para colocar objetos y componentes VEMMI dentro de la DDA (véase la Figura 3). Las coordenadas normalizadas se expresan teóricamente en la gama 0:0 a 1:1. El punto de referencia de origen de la coordenada 0:0 representa la esquina superior izquierda de la DDA, el punto de la coordenada 1:1 representa la esquina inferior derecha de una DDA cuadrada virtual con un lado igual a la unidad. En una DDA no cuadrada, la unidad es igual al lado más grande de la DDA.

En los dispositivos de visualización típicos que utilizan la relación de aspecto común 4:3, el posicionamiento horizontal está en la gama comprendida de 0 a 1 que corresponde a toda la anchura de la DDA, el posicionamiento vertical está en la gama 0 a 0,75 que corresponde a toda la altura de la DDA.

El punto de referencia de origen para colocar un objeto dentro de la DDA es siempre la esquina superior izquierda de la DDA. El punto de referencia de origen para colocar un componente dentro de su objeto progenitor (objeto contenedor) es también la esquina superior izquierda de una DDA virtual asociada a la esquina superior derecha del objeto; esa DDA virtual tiene exactamente el mismo tamaño que la DDA estándar, pero con una traslación de posición hasta el objeto para colocar sus propios componentes.

Los ítems VEMMI se colocan en general implícitamente con respecto a la anchura o la altura del ítem más cercano situado a la izquierda o por encima.

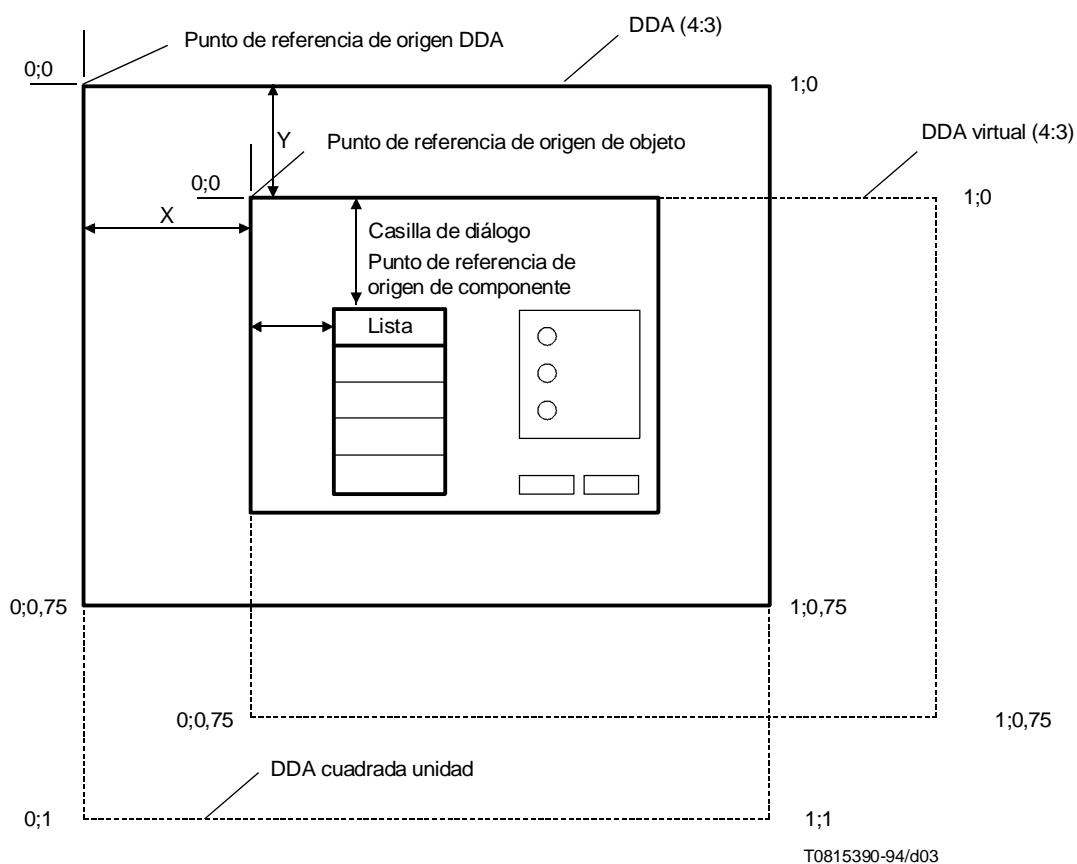


FIGURA 3/T.107
Espacio NDC de posicionamiento VEMMI

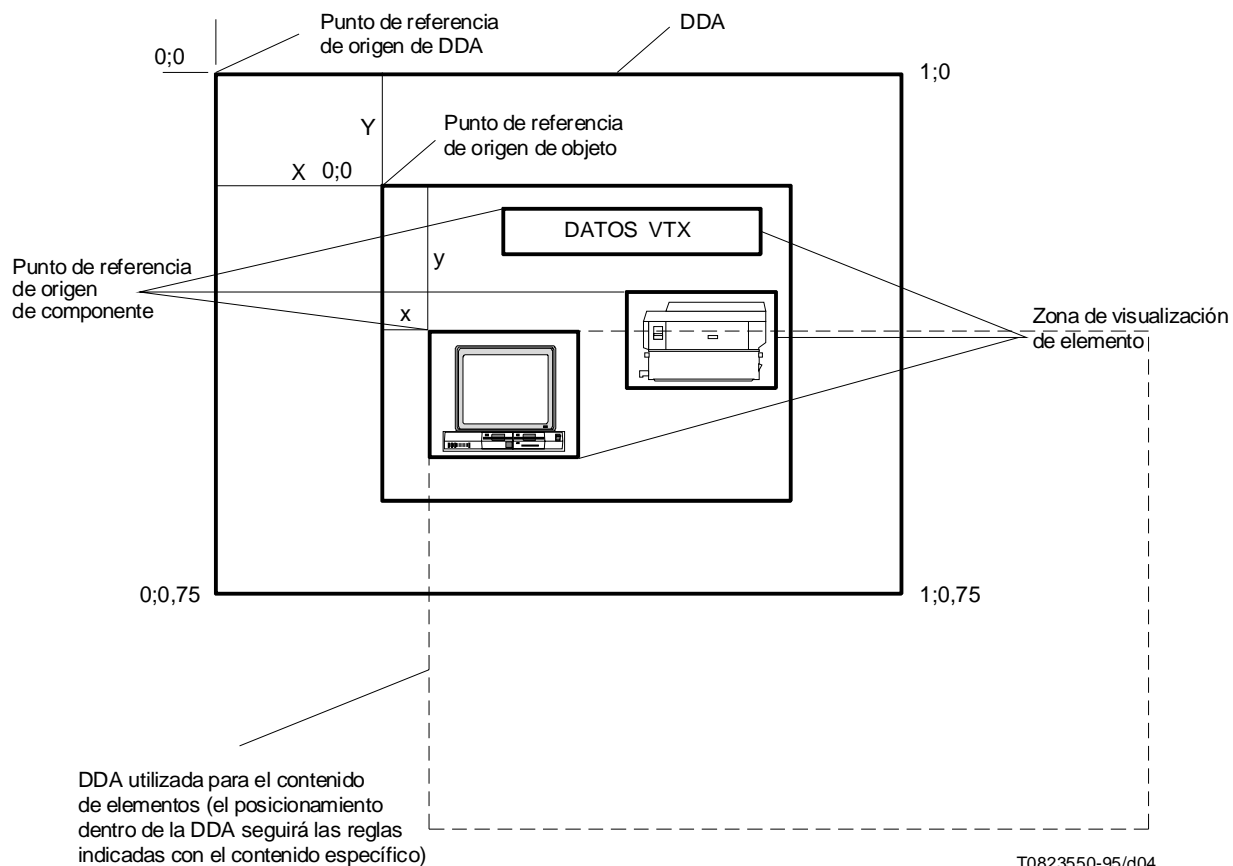
4.6.2 Dimensionamiento

El plano estándar seguirá soportando el sistema de coordenadas utilizado para dimensionar en la plataforma subyacente.

El plano lógico de objetos VEMMI, y los objetos y componentes VEMMI soportarán un sistema de dimensionamiento basado en NDC.

La anchura y la altura de un objeto VEMMI o de un componente VEMMI, expresadas en NDC, se dan con respecto a su punto de referencia de posicionamiento (esquina superior izquierda), véase la Figura 4.

En un elemento VEMMI, el contenido de datos sigue las reglas generales de codificación de su sintaxis de datos pertinente, incluida una posible colocación o posicionamiento de la imagen utilizando el sistema de coordenadas pertinente.



T0823550-95/d04

FIGURA 4/T.107
Sistema de posicionamiento VEMMI

4.7 Estados de elementos VEMMI y parámetros de estado

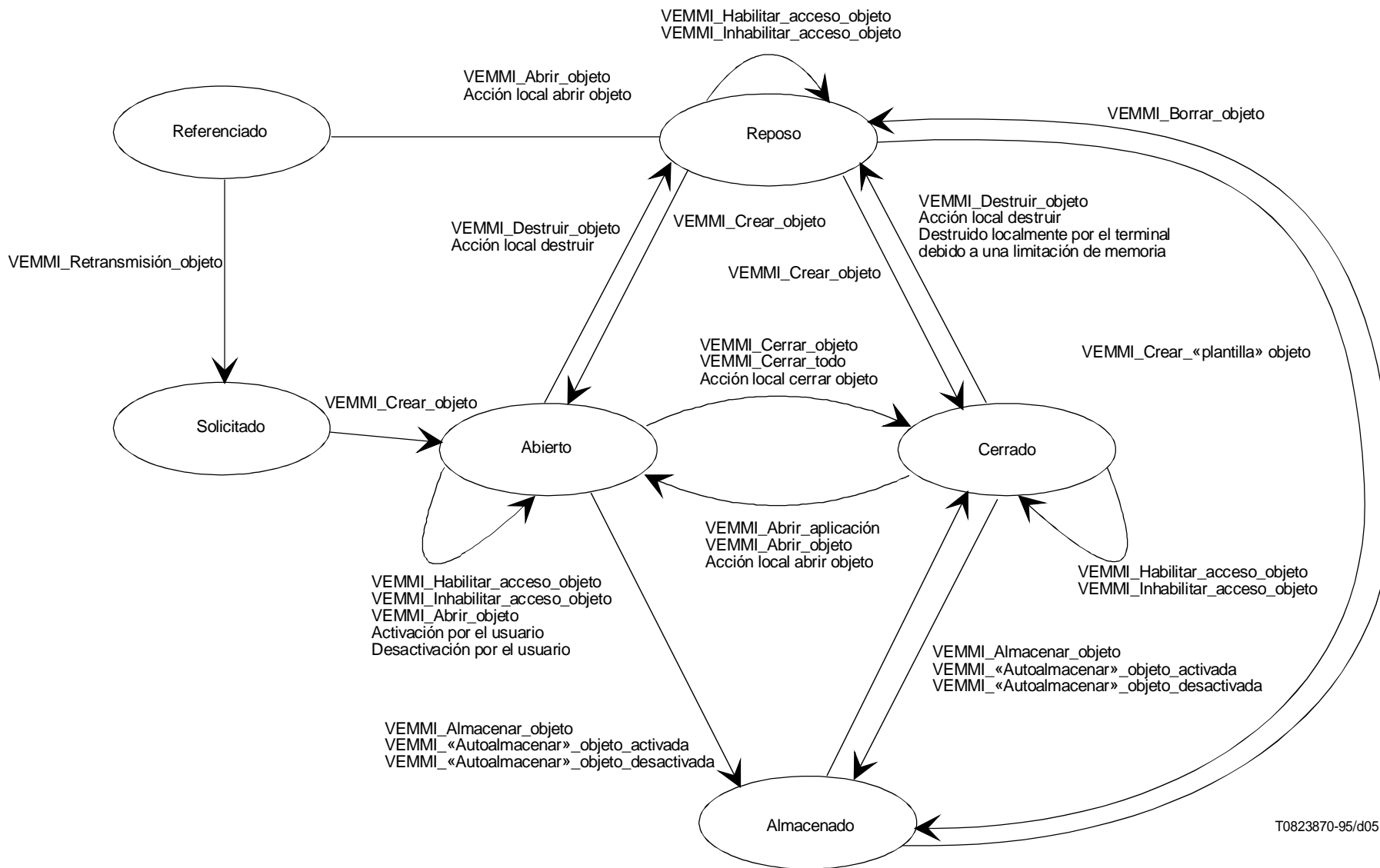
4.7.1 Objeto

Un objeto puede adoptar diferentes estados lógicos con diferentes parámetros de estados. Estos estados y los parámetros de estados pueden ser el resultado de:

- entradas de usuario;
- acciones locales;
- acciones realizadas por el gestor local VEMMI;
- instrucciones de la aplicación VEMMI.

4.7.1.1 Definición de estados de objetos

La Figura 5 muestra las instrucciones que puede emplear la aplicación VEMMI para cambiar el estado de objeto o los parámetros de estado.



T0823870-95/d05

FIGURA 5/T.107
Diagrama de estados para objetos

En general, se definen los siguientes estados lógicos para objetos, con ciertas restricciones aplicables a objetos específicos (véase la cláusula 7):

– *Abierto*

Un objeto que existe en el terminal es visualizado.

La apertura de un objeto es iniciada por la aplicación VEMMI, como estado inicial en el momento de la creación del objeto, o por una instrucción VEMMI directa en cualquier momento durante la aplicación, o por una acción local, que está asociada a un componente y es activada por una interacción con el usuario.

– *Cerrado*

Un objeto que existe en el terminal no es visualizado.

El cierre de un objeto es iniciado por la aplicación VEMMI como estado inicial en el momento de la creación del objeto, o por una instrucción en cualquier momento durante la aplicación, o por una acción local, que está asociada a un componente y es activada por una interacción con el usuario.

– *Referenciado*

Un objeto que no existe en el terminal fue referenciado por una instrucción VEMMI_Abrir_objeto o por la acción local «abrir objeto». El terminal pedirá la retransmisión del objeto especificado mediante la instrucción VEMMI_Retransmisión_objeto.

– *Solicitado*

Después de un intento de abrir un objeto inexistente, el terminal enviará a la aplicación VEMMI una petición de retransmisión de objeto. Mientras el terminal espera la creación del objeto pedido, el objeto se encuentra en el estado solicitado.

– *Almacenado*

Es una copia de un objeto (incluidos todos los componentes), utilizado en una sesión anterior o en la sesión actual abierta, y que ha sido almacenado permanentemente en el terminal (por ejemplo, en el disco duro).

Un objeto almacenado tendrá implícitamente el estado cerrado. Los objetos almacenados se ponen disponibles para operaciones de objeto con la instrucción «abrir aplicación», se comportan entonces como objetos cerrados recibidos.

Si se pide que un objeto previamente almacenado en el terminal sea almacenado nuevamente por la aplicación anfitriona, el objeto nuevo reemplazará al antiguo con todos sus parámetros y estados.

Los objetos almacenados pueden suprimirse con la instrucción VEMMI_Borrar_objeto.

Un objeto que está en el estado cerrado puede ser destruido de la memoria del terminal con una decisión del gestor local VEMMI (por ejemplo, debido a una limitación de la memoria). La aplicación VEMMI puede ser informada de esta destrucción local utilizando la instrucción de error VEMMI. Si el terminal recibe una instrucción VEMMI_Abrir_objeto que se refiere a un objeto que ha sido destruido, o cuando se ejecuta una acción local «abrir objeto» referida al objeto destruido, el terminal pedirá la retransmisión del objeto desde la aplicación mediante la instrucción VEMMI_Retransmisión_objeto. Entonces, la aplicación VEMMI volverá a crear el objeto solicitado con la instrucción VEMMI_Crear_objeto, aplicando su estado vigente y sus parámetros de estado vigentes dentro de la aplicación VEMMI. Una instrucción VEMMI_Crear_objeto resultante de una petición VEMMI_Retransmisión_objeto creará el objeto siempre en el estado abierto.

4.7.1.2 Definición de parámetros de estados de objeto

En general, se definen los siguientes parámetros de estados para objetos, con algunas restricciones aplicables a objetos específicos (véase la cláusula 7):

– *Activo*

Un objeto tiene en ese momento el enfoque de entrada. Las entradas de usuario se refieren siempre al objeto activo. El objeto activo está por encima de la DDA. Si el objeto activo es inaccesible, el usuario no podrá interactuar con él.

La gestión del estado activo se describe en 4.10.1.

– *Inactivo*

Lo contrario de activo.

- *Accesible*

El usuario puede interactuar con el objeto.

La habilitación de la interacción con un objeto es iniciada por la aplicación VEMMI, como estado inicial en el momento de la creación del objeto, por una instrucción VEMMI en cualquier momento de la aplicación o por una acción local, que está asociada con un componente y es activada por una interacción con el usuario.

- *Inaccesible*

Lo contrario de accesible.

El parámetro de estado activo/inactivo se aplica sólo a componentes que están en el estado abierto. Para un componente abierto puede existir cualquier combinación de parámetros de estado activo/inactivo y accesible/inaccesible.

4.7.2 Componente

4.7.2.1 Definición de estados de componente

Un componente puede adoptar diferentes estados lógicos con diferentes parámetros de estado. Estos estados y parámetros de estado pueden ser el resultado de:

- entradas de usuario;
- acciones locales;
- instrucciones de la aplicación VEMMI.

La Figura 6 muestra los estados posibles de los componentes. Los estados y los parámetros de estado de los componentes pueden cambiarse durante las aplicaciones mediante la instrucción VEMMI_Modificar_componente.

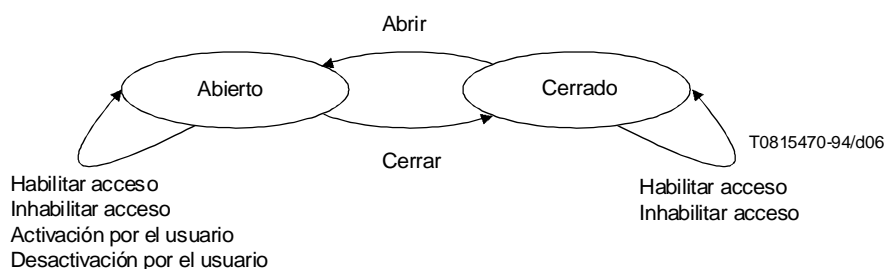


FIGURA 6/T.107

Diagramas de estados para componentes

En general, se definen los siguientes estados lógicos para componentes, con algunas restricciones aplicables a componentes específicos (véase la cláusula 7):

- *Abierto*

Un componente que existe dentro de un objeto es visualizado.

La apertura de un componente es iniciada por la aplicación VEMMI, como estado inicial en el momento de la creación del objeto, por una instrucción VEMMI_Modificar_componente, en cualquier momento durante la aplicación, o por una acción local que está asociada con un componente y es activada por una interacción con el usuario.

- *Cerrado*

Un componente que existe en el terminal no es visualizado.

El cierre de un componente es iniciado por la aplicación VEMMI, como estado inicial en el momento de la creación del objeto, o por una instrucción VEMMI_Modificar_componente, en cualquier momento durante la aplicación, o por una acción local que está asociada con un componente y es activada por una interacción con el usuario.

4.7.2.2 Definición de parámetros de estados de componente

En general, se definen los siguientes parámetros de estados para componentes, con algunas restricciones aplicables a componentes específicos (véase la cláusula 7):

– *Activo*

Un componente tiene en ese momento el enfoque de entrada. Las entradas de usuarios se refieren siempre al componente activo dentro del objeto activo. Si el componente activo está inaccesible, el usuario no podrá interactuar con él.

La gestión del estado activo se describe en 4.10.1.

– *Inactivo*

Lo contrario de activo.

– *Accesible*

Un componente puede ser activado por el usuario, si forma parte del objeto activo vigente.

La habilitación de la interacción de un usuario con un componente es iniciada por la aplicación VEMMI, como estado inicial en el momento de la creación del objeto, por una instrucción VEMMI_Modificar_componente, en cualquier momento durante la aplicación, o por una acción local que está asociada con un componente y es activada por una interacción con el usuario.

– *Inaccesible*

Lo contrario de accesible.

4.8 Gestión de acciones locales

Las instrucciones VEMMI son emitidas generalmente por la aplicación VEMMI para cambiar el estado o los parámetros de estado de un objeto o componente. Sólo un número muy limitado de cambios de objeto, o de estados de componente, o de sus parámetros puede ser iniciado por el terminal.

Para que la aplicación VEMMI pueda interactuar con el usuario, el terminal proporcionará la capacidad de informar los valores de entrada de usuario de los componentes, su acceso o su validación.

El resultado de estas interacciones es que los cambios de objeto o de estados de componente, que son el resultado de las entradas de usuario, serían retardadas, porque la entrada de usuario sería enviada primero a la aplicación VEMMI, y después la aplicación podría reaccionar con una instrucción VEMMI para cambiar el estado. Este procedimiento pudiera causar tiempos de respuesta inadmisibles en redes de baja velocidad. Para mejorar los tiempos de respuesta, se ha definido un conjunto de acciones locales.

Las acciones locales forman parte de las definiciones de componentes y están almacenadas en el terminal en el momento de la creación del objeto. Cada acción local puede constar de una lista de «instrucciones específicas de elementos», una lista de «instrucciones de informe» y una lista de «instrucciones generales». Se definen las instrucciones siguientes:

a) *Instrucciones específicas de elementos*

- abrir componentes del objeto progenitor (Parámetro: lista de CIN);
- cerrar componentes del objeto progenitor (Parámetro: lista de CIN);
- abrir objetos (Parámetro: lista de OIN);
- cerrar objetos (Parámetro: lista de OIN);
- cambiar parámetro de estado de componente de un componente en el objeto progenitor a inaccesible (Parámetro: lista de CIN);
- cambiar parámetro de estado de componente de un componente en el objeto progenitor a accesible (Parámetro: lista de CIN);
- destruir objetos (Parámetro: lista de OIN);
- abrir objeto bloqueante (Parámetro: OIN).

b) *Instrucciones de informe*

- comunicar OIN y CIN;
- comunicar los valores actuales del componente;

- comunicar los valores de todos los componentes del objeto;
 - comunicar todos los valores de todos los componentes del objeto progenitor que hayan cambiado (mediante una `VEMMI_Modificar_componente`) desde el último informe o desde la creación del objeto si éste es el primer informe.
- c) *Instrucciones generales*
- bloqueo de usuario;
 - restablecer los valores iniciales de todos los componentes del objeto progenitor. Esta instrucción se aplicará únicamente si el objeto se creó con el atributo «StoreInitialValue» puesto a verdadero.

Estas instrucciones son parte de la definición de objeto, y pueden aparecer en la acción local de un componente en cualquier orden posible. El orden de aparición será igual al de su ejecución. Hay dos eventos generadores específicos que provocan la ejecución de acciones locales:

- activación de un componente;
- validación de un componente.

Puede utilizarse una lista de «instrucciones específicas de elementos», una lista de «instrucciones de informe» y una lista de «instrucciones generales» para especificar una acción local de un componente. Ninguna instrucción específica es obligatoria para una acción local.

Hay dos eventos generadores (*trigger events*) específicos que provocan la ejecución de acciones locales:

- activación de un componente;
- validación de un componente.

Puede asociarse una acción local a cada uno de estos eventos generadores.

4.9 Consideraciones sobre memoria

Se supone que los terminales VEMMI tienen capacidades de memoria, correctamente dimensionadas para proporcionar un servicio VEMMI satisfactorio, en estrecha relación con su tratamiento del contenido del tipo de datos.

Cuando no hay suficiente memoria para ejecutar una instrucción VEMMI o una acción local, el gestor local puede tomar la decisión de destruir algunos objetos cerrados. Este mecanismo depende del terminal.

El terminal VEMMI enviará un mensaje `VEMMI_Error` (fuera de memoria) a la aplicación cuando no tiene la posibilidad de liberar suficiente memoria para procesar una descripción de objeto recibido con una instrucción `VEMMI_Crear_objeto`. En ese caso, puede ignorar otras instrucciones `VEMMI_Crear_objeto`, mientras la aplicación VEMMI no haya liberado una parte importante de memoria utilizando las instrucciones `VEMMI_Cerrar_objeto` o `VEMMI_Destruir_objeto`. La aplicación VEMMI puede también decidir desconectar la línea si la aplicación no puede ser ejecutada correctamente por falta de memoria en el terminal; sin embargo, el usuario será siempre informado de esa decisión por medio de un mensaje en una casilla de mensajes.

4.10 Reglas comunes para el tratamiento de objetos

4.10.1 Estado activo y gestión de enfoque

Un objeto VEMMI puede volverse activo como resultado de:

- a) apertura por la aplicación VEMMI;
- b) acceso de usuario;
- c) una acción local con la instrucción específica de elemento «abrir objeto»;
- d) cierre del objeto activo. El objeto abierto activado más recientemente se convierte en el nuevo objeto activo.

Se dará el estado activo al último objeto VEMMI accesible abierto, hasta un acceso de usuario a un objeto VEMMI diferente. Cuando un usuario accede a un objeto VEMMI abierto accesible, el gestor local dará el estado activo a este objeto VEMMI. El estado activo es tratado por el gestor local hasta una nueva acción de apertura o reapertura de un objeto VEMMI precedente de la aplicación VEMMI. Habrá siempre un solo objeto activo en la DDA. El objeto activo puede ser inaccesible.

Si los objetos utilizan una zona común en la DDA, la activación de objetos sucesivos conduce a una superposición parcial o total de objetos. El objeto activo será siempre totalmente visible al usuario exactamente después de su activación. El objeto activo anterior está lógicamente debajo del objeto activo. Esta regla se aplicará a todos los objetos anteriormente activos.

Cuando está en el estado activo, el objeto VEMMI y/o uno de sus componentes accesibles recibirá su enfoque del gestor local. El estado activo del objeto VEMMI y/o de los componentes VEMMI será indicado visualmente de manera clara al usuario. Dentro del objeto VEMMI activo, sólo un componente VEMMI accesible recibirá el estado activo y el enfoque.

La primera vez que un objeto es visualizado y después activado, la aplicación VEMMI puede especificar el componente al que se da el enfoque dentro de la descripción de objeto. Si no hay un componente especificado, o si el componente especificado no está abierto, el terminal dará el enfoque al primer componente creado abierto y activo. En otros casos, la manera de dar el enfoque a uno de los componentes cuando se activa un objeto depende del terminal (véase la regla anterior, memorización del último componente accedido, el componente de acceso de usuario más cercano al objeto progenitor).

4.10.2 Comportamiento del modo modal

Para un objeto activo modal, toda interacción con otro objeto VEMMI de la misma aplicación VEMMI está prohibida por el terminal, y cualquier intento se le comunica al usuario (pitido, etc.).

Se ejecutan todas las instrucciones recibidas del anfitrión y todas las acciones locales resultantes de las interacciones de usuario con los componentes del objeto modal.

4.10.3 Consideraciones relativas al tamaño y recorte

El contenido de datos de un elemento VEMMI será enviado con dimensiones y una posible posición compatibles con la descripción de tamaño de su contenedor (objeto o componente). Un terminal VEMMI recortará un contenido de datos que no concuerde con las dimensiones globales del contenedor.

Para el contenido de datos del mapa de bits, se aplican reglas específicas que se dan con la descripción de los elementos.

4.11 Almacenamiento de objetos locales

Una aplicación anfitriona tiene la posibilidad de pedir la carga de objetos localmente almacenados en el terminal. Ese conjunto de objetos es específico de la aplicación y contiene usualmente aquellas partes que no se modifican durante un periodo largo, por lo que resulta más eficaz no transmitirlos al terminal en cada sesión de diálogo. Los objetos cargados se procesan igual que si hubieran sido recibidos durante la sesión presente. Ante una petición del anfitrión, pueden restablecerse de nuevo para mantenerlos actualizados. El almacenamiento de objetos no significa que se ha sacado a los objetos de la aplicación de terminal, pues lo que se almacena es una copia de los objetos. Un conjunto de objetos se identifica mediante el identificador de aplicación.

La indicación de tiempo (timestamp, en inglés) se recibe del anfitrión al comienzo de la aplicación. Se almacena con los objetos y se reutiliza más adelante en otras sesiones. El terminal nunca calcula el valor de la indicación de tiempo, sólo la utiliza para encontrar, por comparación, objetos de fechas anteriores. Cuando un objeto se almacena localmente en el terminal, llevará asociada una indicación de tiempo. El valor de la indicación de tiempo se calculará así:

- si el objeto fue creado por el anfitrión (con VEMMI_Crear_objeto) durante la sesión actual, o si cualquier componente del objeto fue modificado (con VEMMI_Modificar_componente) durante la sesión actual, el objeto se almacenará con el valor de indicación de tiempo determinado por la instrucción VEMMI_Abrir_aplicación correspondiente a la aplicación actual;
- en los demás casos, la indicación de tiempo del objeto no sufrirá modificaciones. (El objeto fue cargado sin cambios desde el almacenamiento local durante esta sesión.)

Además, pueden utilizarse dos atributos, que forman parte de la definición de objeto, para pedir el almacenamiento de objetos, ya sea inmediatamente después de la creación de los objetos (plantilla), o al terminar la aplicación (autoalmacenamiento).

Un conjunto de objetos almacenado contiene lo siguiente:

- el identificador de aplicación;
- los objetos, incluidos sus componentes, y la indicación de tiempo asociada;

- los ficheros referenciados (si un fichero referenciado no está disponible en el terminal, el objeto se considera incompleto y no será almacenado);
- la tabla de colores cargada correspondiente.

El formato de almacenamiento depende del terminal porque los objetos sólo los carga y almacena el mismo terminal.

La implementación de terminal debe proporcionar al usuario los medios adecuados para gestionar el almacenamiento local y el usuario podría limitar su tamaño máximo.

4.12 Nombres de directorios simbólicos

A fin de soportar la aplicación anfitriona de localizar los ficheros adecuados, se definen nombres de directorios simbólicos. Pueden utilizarse como parte del parámetro «nombre de fichero» (*filename*). El terminal los hace corresponder con nombres de ficheros de directorios reales. En el ejemplo que sigue una aplicación anfitriona utiliza un nombre de directorio simbólico para identificar un fichero de capas de bits.

VEMMI_Crear_objeto (OIN = 7, ..., nombre de fichero = «\$CD\ARTISTS\DANCER\LISA.BMP»; tipo de fichero de imagen = BMP).

La cadena «\$CD» identifica la unidad de disco CD-ROM. El terminal sustituye este nombre simbólico por la identificación de la unidad de disco real. La identificación de fichero completo podría ser «H:\ARTISTS\DANCER\LISA.BMP».

Se definen los siguientes nombres simbólicos; la aplicación puede utilizar otros nombres simbólicos:

Nombre	Significado
\$APL	Directorio progenitor de las aplicaciones almacenadas en el terminal
\$FTD	Directorio progenitor de los ficheros telecargados
\$CD	La unidad de disco indica el nombre de directorio de una unidad de disco CD-ROM

Estos nombres simbólicos pueden utilizarse en el parámetro «nombre de fichero» de las instrucciones VEMMI. El nombre simbólico aparecerá sólo una vez en el parámetro nombre de fichero. Para separar el nombre simbólico de los subdirectorios adicionales y del nombre de fichero, debe utilizarse el carácter «\», código 5/12.

4.13 Reglas específicas para terminales dedicados

Todas las primitivas de servicio, incluido sus parámetros, serán también entendidas por un terminal dedicado. Con respecto a la ejecución de las instrucciones relativas a una funcionalidad específica, un terminal dedicado puede aplicar mecanismos de degradación progresiva.

La siguiente funcionalidad puede ser degradada progresivamente por un terminal dedicado.

- definición de color;
- atributos del texto (tipos de caracteres, colores, atributos de tipos de caracteres);
- almacenamiento local;
- posicionamiento NDC y dimensionamiento.

Para terminales VEMMI dedicados aptos únicamente para tratar contenido de datos videotex de tipo caracteres (alfamosaicos, DRCS, perfil fotográfico 1), los objetos y componentes VEMMI pueden colocarse físicamente y dimensionarse a la posición de carácter de visualización correspondiente más próxima. Por consiguiente, para evitar los efectos secundarios entre objetos VEMMI, las posiciones de objetos NDC deben expresarse en un múltiplo directo de las posiciones de carácter para aplicaciones VEMMI que utilicen sólo este tipo de contenido de datos videotex.

Para terminales VEMMI dedicados sólo aptos para tratar contenido de datos videotex por caracteres, puede utilizarse VEMMI con modo 80 caracteres utilizando el elemento de datos de presentación videotex (VPDE) SECUENCIA DE CONMUTACIÓN o DEFINIR FORMATO adecuado para conmutar a la sintaxis de datos deseada. El modo 80 caracteres no es obligatorio en un terminal VEMMI.

Los terminales VEMMI dedicados reconocerán, pero no necesariamente procesarán, la totalidad o parte de los atributos de textos de alta calidad VEMMI que no son capaces de visualizar. Estos terminales pueden entonces ignorar los atributos de texto y visualizar texto utilizando el tipo de caracteres del sistema, en el color de sistema disponible, sobre el fondo del sistema.

Los terminales VEMMI dedicados soportarán los diferentes tipos de objetos VEMMI, con posibles restricciones aplicables a los objetos operativos. Como no es obligatorio el sistema de gestión de ficheros, pueden gestionar los ficheros que reciben en la instrucción VEMMI_Transferencia_recurso en su propia representación interna.

Un terminal dedicado debe soportar el mínimo número de tipos de datos de núcleo (kernel) definidos en el anexo C. Sin embargo, cuando se utilizan en un servicio de recuperación en el que el modo estándar es el modo videotex y cuando el terminal no soporta la sintaxis fotográfica videotex (por la que hay que preguntar utilizando el TFI), no es obligatorio el soporte del JPEG y GIF. Cuando el terminal soporta la sintaxis fotográfica videotex, es obligatorio el soporte JPEG; se recomienda el soporte de GIF. En las mismas condiciones de servicio, cuando un terminal no soporta la sintaxis audio videotex, no es obligatorio el soporte de WAVE y MIDI. Cuando el terminal soporta la sintaxis audio videotex, el soporte de WAVE es obligatorio; se recomienda el soporte de MIDI.

Los terminales dedicados no necesitan soportar la elastización de los datos de mapas de bits para llenar la zona de visualización de elementos. Si el atributo de presentación de un mapa de bits se pone a «elástico», el terminal puede presentar el mapa de bits enlosado.

5 Descripción de servicio

Esta cláusula describe las características de servicio ofrecidas por VEMMI.

El servicio se define entre un terminal y una aplicación distante. No se hacen hipótesis sobre la manera en que el terminal videotex accede al servicio (RTPC/RPDCP/RDSI) ni la manera en que se establece la conexión.

Todos los elementos de servicio VEMMI o instrucciones VEMMI serán emitidos solamente por una conexión de red establecida. Antes de enviar la primera instrucción VEMMI, la aplicación VEMMI debe emitir una petición de identificador de facilidad de terminal (TFI) para asegurarse de que el terminal soporta la norma VEMMI.

Todos los elementos del servicio VEMMI son obligatorios para un terminal VEMMI. Un terminal VEMMI entenderá todos los elementos de servicio VEMMI, incluidos todos sus parámetros.

Los servicios VEMMI se dividen en dos grupos:

- servicios iniciados por la aplicación VEMMI;
- servicios iniciados por el terminal VEMMI.

Una segunda fuente de instrucciones para un terminal VEMMI es el objeto metacódigo. Véase el Cuadro 2.

Los siguientes servicios se confirman explícitamente:

- VEMMI_Identificar_cap_term se confirma con VEMMI_Resp_identificar_cap_term;
- VEMMI_Abrir_aplicación se confirma con VEMMI_Resp_abrir_aplicación;
- VEMMI_Almacenar_objetos se confirma con VEMMI_Resp_almacenar_objetos.

No hay confirmación explícita en ninguno de los elementos de servicio.

Todos los servicios VEMMI, excepto VEMMI_Objeto, que se refiere a un objeto inexistente, serán ignorados por el terminal VEMMI. Si una VEMMI_Objeto se refiere a un objeto inexistente, el terminal solicitará su retransmisión.

NOTA – En la subcláusula siguiente se describen todos los servicios VEMMI, pero sólo se enumeran los parámetros necesarios para entender el servicio. Para una lista completa de los parámetros, véase la cláusula 8.

CUADRO 2/T.107

Servicios VEMMI

Servicios VEMMI	Iniciados por aplicación/terminal	Función
VEMMI_Fijar_modo_traducción	Aplicación/terminal	Fijar el modo de traducción para datos VEMMI
VEMMI_Abrir	Aplicación	Cambiar a modo VEMMI y reiniciar
VEMMI_Cerrar	Aplicación	Cambiar a modo estándar y reiniciar
VEMMI_Suspender	Aplicación	Cambiar a modo estándar
VEMMI_Reanudar	Aplicación	Cambiar a modo VEMMI
VEMMI_Identificar_cap_term	Aplicación	Petición de capacidades de terminal
VEMMI_Resp_identificar_cap_term	Terminal	Respuesta a la petición «identificar capacidad de terminal»
VEMMI_Fijar_opciones	Aplicación	Fijar las opciones en el terminal
VEMMI_Crear_objeto	Aplicación	Definición de objeto
VEMMI_Abrir_objeto	Aplicación	Visualizar el objeto
VEMMI_Abrir_objeto_bloqueante	Aplicación	Abrir el objeto
VEMMI_Cerrar_objeto	Aplicación	Borrar un objeto de la pantalla
VEMMI_Cerrar_todos	Aplicación	Borrar todos los objetos de la pantalla
VEMMI_Destruir_objeto	Aplicación	Borrar el objeto de la memoria del terminal
VEMMI_Inhabilitar_acceso_objeto	Aplicación	No se permite el acceso de usuario
VEMMI_Habilitar_acceso_objeto	Aplicación	Se permite el acceso de usuario
VEMMI_Modificar_componente	Aplicación	Modificar las características del componente
VEMMI_Cambiar_ubicación_objeto	Aplicación	Definir una nueva posición del objeto
VEMMI_Cargar_tabla_col	Aplicación	Cargar una tabla de colores
VEMMI_Restablecer_tabla_col	Aplicación	Restablecer una tabla de colores
VEMMI_Abrir_aplicación	Aplicación	Abrir una aplicación
VEMMI_Resp_abrir_aplicación	Terminal	Respuesta a una petición «abrir aplicación»
VEMMI_Borrar_objeto_antiguado	Aplicación	Borrar objetos antiguos
VEMMI_Almacenar_objetos	Aplicación	Almacenar un conjunto de objetos
VEMMI_Resp_almacenar_objetos	Terminal	Respuesta a una petición de almacenar objetos
VEMMI_Borrar_objetos	Aplicación	Borrar objetos del archivo local
VEMMI_Bloquear_usuario	Aplicación	No se permiten entradas de usuario
VEMMI_Desbloquear_usuario	Aplicación	Se permiten entradas de usuario
VEMMI_Retransmisión_objeto	Terminal	Petición de retransmisión de un objeto
VEMMI_Datos_usuario	Terminal	Datos de usuario en modo VEMMI
VEMMI_Error	Terminal	Condición o situación de error
VEMMI_Transferencia_recurso	Aplicación	Transferir recurso
VEMMI_Abortar_transferencia_recurso	Terminal	Abortar transferencia de recurso

5.1 Elementos de servicio iniciados por la aplicación VEMMI y el terminal

Si es emitido por la aplicación VEMMI, este elemento de servicio se utilizará para indicar que los datos VEMMI enviados desde la aplicación al terminal se transcodifican utilizando un modo de traducción especificado.

Si es emitido por el terminal VEMMI, este elemento de servicio se utilizará para indicar que los datos VEMMI enviados desde el terminal a la aplicación se transcodifican utilizando un modo de traducción especificado.

Dado que un terminal VEMMI suele comenzar la operación, por defecto, en un entorno de 7 o de 8 bits, no será obligatorio que una aplicación VEMMI envíe como primera instrucción VEMMI en una sesión VEMMI una VEMMI_Fijar_modos_traducción para conmutar el terminal a un modo de traducción deseado. Si la aplicación no puede estar segura de que todos los terminales que están conectando operan en un entorno específico, debe enviar una VEMMI_Fijar_modos_traducción para asegurarse de que el terminal interpretará correctamente las instrucciones VEMMI. Véase el Cuadro 3.

La VEMMI_Fijar_modos_traducción suele ser la primera instrucción VEMMI de una sesión VEMMI, pero puede utilizarse en cualquier momento durante una sesión VEMMI.

NOTA – El modo de traducción puede ser diferente en los dos sentidos.

CUADRO 3/T.107

Parámetros

Parámetros	Descripción
Modo de traducción	Un byte que especifica el modo de traducción deseado

El parámetro modo de traducción (translationMode) puede tener los siguientes valores:

- 2/0: no hay traducción (transparente);
- 2/1: codificación 3 en 4 (véase la Recomendación T.101 [4]);
- 2/2: codificación de 7 bits (véase la Recomendación T.101 [4]).

5.2 Elementos de servicio iniciados por la aplicación VEMMI

5.2.1 VEMMI_Abrir

Este elemento de servicio se utilizará para conmutar el terminal VEMMI al modo de funcionamiento VEMMI. El mecanismo de conmutación se define en 4.4.3. En el modo de funcionamiento VEMMI, se aplicará el mecanismo descrito en 4.4.2.

Además de cambiar al modo de operación VEMMI, el elemento de servicio efectuará las reiniciaciones siguientes:

- cerrar una aplicación abierta;
- destruir todos los objetos del conjunto de objetos que están en el estado «abierto» o «cerrado»;
- borrar de la pantalla todos los objetos abiertos;
- liberar todos los números de identificación de los objetos de recurso;
- fijar la tabla de colores a los valores predefinidos;

Los objetos almacenados en un archivo local no son afectados.

Si esta instrucción se recibe en modo VEMMI, no es necesaria ninguna conmutación, y sólo se efectuarán las reiniciaciones. Véase el Cuadro 4.

CUADRO 4/T.107

Parámetros

Parámetros	Descripción
Versión	Identificador de la versión VEMMI
Modo privado	Parámetro privado que se puede usar en algunos servicios videotex «por páginas». Este modo no se describe en esta Recomendación

5.2.2 VEMMI_Cerrar

Este elemento de servicio se utilizará para conmutar el terminal VEMMI al modo videotex estándar. El mecanismo de conmutación se define en 4.4.3. En el modo de operación estándar, se aplicará el mecanismo descrito en 4.4.1.

Además de la conmutación al modo estándar, el elemento de servicio efectuará las reiniciaciones definidas en 5.2.1. Véase el Cuadro 5.

CUADRO 5/T.107

Parámetros

Parámetros	Descripción
Ninguno	

5.2.3 VEMMI_Reanudar

Este elemento de servicio se utilizará para conmutar el terminal VEMMI al modo de operación VEMMI. Los parámetros de este elemento de servicio son los mismos que para VEMMI_Abrir (véase 5.2.1). Véase el Cuadro 6.

CUADRO 6/T.107

Parámetros

Parámetros	Descripción
Versión	Identificador de la versión VEMMI
Modo privado	Parámetro privado que se puede usar en algunos servicios videotex «por páginas». Este modo no se describe en esta Recomendación

5.2.4 VEMMI_Suspender

Este elemento de servicio se utilizará para conmutar el terminal VEMMI al modo estándar. No se efectuará ninguna reiniciación. Véase el Cuadro 7.

CUADRO 7/T.107

Parámetros

Parámetros	Descripción
Ninguno	

5.2.5 VEMMI_Identificar_cap_term

Este elemento de servicio se utilizará para solicitar la identificación de las capacidades de terminal. Véase el Cuadro 8.

Puede obtenerse la información siguiente:

- identificación de la versión VEMMI;
- soporte de almacenamiento local (soportado/no soportado);
- lista de idiomas de usuarios preferidos (cadena);
- tipo de sistema (por ejemplo: IBM compatible, Macintosh, ...) sistema operativo (por ejemplo: MS-DOS, OS2, Microsoft Windows, ...), versión de sistema operativo (por ejemplo: 3.1) (cadena);
- una lista de los tipos de datos soportados (referencia al catálogo de identificación de codificación de contenido VEMMI).

Si el parámetro información deseada no está presente, el terminal responderá con una lista completa que identifica todas sus capacidades.

CUADRO 8/T.107

Parámetros

Parámetros	Descripción
Información deseada	Tipo de información que se solicita al terminal

5.2.6 VEMMI_Fijar_opciones

Este elemento de servicio se utilizará para fijar las opciones en el terminal. Véase el Cuadro 9.

Pueden fijarse las opciones siguientes:

- si el terminal soporta diferentes tipos de codificación de texto, puede seleccionar la norma de codificación deseada.

CUADRO 9/T.107

Parámetros

Parámetros	Descripción
Norma de texto	Referencia a una norma de codificación de texto del catálogo VEMMI de identificación de la codificación de contenido que se utilizará al comienzo de la codificación de los tipos de datos cadena y carácter

5.2.7 VEMMI_Crear_objeto

Este elemento de servicio se utilizará para definir un objeto VEMMI y sus componentes en el terminal VEMMI. El terminal VEMMI almacenará el objeto y aplicará el estado del objeto y los parámetros de estado indicados en su definición.

Los OIN serán únicos cada vez en una aplicación. La creación de un objeto con un OIN ya existente causará la destrucción del objeto definido previamente.

Los CIN serán únicos dentro de un objeto. La definición de un componente con un CIN ya existente causará la destrucción del componente definido previamente. Véase el Cuadro 10.

CUADRO 10/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	
Plantilla	Si una aplicación está abierta, el objeto se almacenará inmediatamente después de la creación
Autoalmacenamiento	Si una aplicación está abierta, el objeto se almacenará al recibir una VEMMI_Abrir, una VEMMI_Cerrar o una VEMMI_Abrir_aplicación

5.2.8 VEMMI_Abrir_objeto

Este elemento de servicio se utilizará para poner un objeto en el estado abierto en el terminal VEMMI. El terminal visualizará inmediatamente el objeto especificado. Si el objeto no está presente en el terminal, pedirá el objeto enviando una Petición VEMMI_Retransmisión_de_objeto a la aplicación VEMMI con el OIN especificado; en este caso se usará la regla A descrita en 5.3.2. La aplicación creará el objeto de nuevo, aplicando su estado y parámetros de estado vigentes.

Una instrucción VEMMI_Abrir_objeto, que se refiere a un objeto ya abierto, causa la activación de este objeto.

Si se aplica VEMMI_Abrir_objeto a un objeto recurso, el terminal verifica si el objeto está disponible. Si no lo está, lo pedirá. No se ejecutará ninguna otra acción.

Si se aplica VEMMI_Abrir_objeto a un objeto metacódigo, el terminal ejecuta el contenido de objeto.

Si se aplica VEMMI_Abrir_objeto a un objeto operativo, el terminal ejecuta el programa. Si no está presente, será solicitado. Véase el Cuadro 11.

CUADRO 11/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	

5.2.9 VEMMI_Abrir_objeto_bloqueante

Este elemento de servicio hace lo mismo que VEMMI_Abrir_objeto, salvo que se empleará la regla B del caso de retransmisión descrita en 5.3.2.

5.2.10 VEMMI_Cerrar_objeto

Este elemento de servicio se utilizará para cerrar un objeto VEMMI en el terminal VEMMI. El terminal VEMMI borra el objeto indicado de la pantalla, pero lo mantiene en su memoria. No se cambiarán los parámetros de estado y valores vigentes de los componentes.

Al recibir una indicación VEMMI_Cerrar_objeto que se refiere a un objeto activo, el nuevo objeto activo en el terminal VEMMI será el objeto abierto más recientemente activo.

Con objetos que no son visualizables o que ya están en un estado cerrado, VEMMI_Cerrar_objeto no tiene ningún efecto. Véase el Cuadro 12.

CUADRO 12/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	

5.2.11 VEMMI_Cerrar_todo

Este elemento de servicio se utilizará para cerrar todos los objetos VEMMI en el terminal VEMMI. Un elemento de servicio VEMMI_Cerrar_todo, no cambiará el modo de operación vigente del terminal. Véase el Cuadro 13.

CUADRO 13/T.107

Parámetros

Parámetros	Descripción
Ninguno	

5.2.12 VEMMI_Destruir_objeto

Este elemento de servicio se utilizará para suprimir un objeto VEMMI del conjunto de objetos que están en el estado «abierto» o «cerrado». El terminal suprimirá de su memoria todos los datos pertenecientes a este objeto. Si se aplica VEMMI_Destruir_objeto a un objeto abierto, éste será cerrado por el gestor local y después destruido. Si el objeto estaba activo, el nuevo objeto activo será el objeto abierto más recientemente activo. Véase el Cuadro 14.

CUADRO 14/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	

5.2.13 VEMMI_Inhabilitar_acceso_objeto

Este elemento de servicio se utilizará para restringir el acceso de usuario a un objeto en el terminal VEMMI. El acceso será restringido hasta que el terminal VEMMI reciba un elemento VEMMI_Habilitar_acceso_objeto relativo al mismo objeto.

Si el objeto estaba activo, el nuevo objeto activo será el objeto abierto que estuvo más recientemente activo en el terminal. Una indicación VEMMI_Inhabilitar_acceso de objeto relativa a un objeto activo interrumpirá la interacción de usuario con este objeto. Véase el Cuadro 15.

CUADRO 15/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	

5.2.14 VEMMI_Habilitar_acceso_objeto

Este elemento de servicio se utilizará para que el usuario pueda acceder a un objeto en el terminal VEMMI. Se permitirá el acceso hasta que el terminal VEMMI reciba un elemento de servicio VEMMI_Inhabilitar_acceso de objeto en relación con el mismo objeto.

Una indicación VEMMI_Habilitar_acceso_objeto recibida en el terminal VEMMI y relacionada con un objeto inactivo no cambiará este parámetro de estado. Véase el Cuadro 16.

CUADRO 16/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	

5.2.15 VEMMI_Modificar_componente

Este elemento de servicio se utilizará para modificar uno o varios componentes VEMMI pertenecientes a un objeto VEMMI creado en el terminal VEMMI. Las posibles modificaciones de componentes VEMMI se definen en los cuadros de definición de componentes de 9.6.

Si se aplica una instrucción VEMMI_Modificar_componente a un objeto abierto, tendrá un efecto visual inmediato en el terminal VEMMI si la parte modificada es visible.

Una instrucción VEMMI_Modificar_componente no cambiará el parámetro de estado activo/inactivo del objeto en el terminal VEMMI. Véase el Cuadro 17.

CUADRO 17/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	
Descripción de modificación	

5.2.16 VEMMI_Cambiar_ubicación_objeto

Este elemento de servicio se utilizará solamente para cambiar la posición de un objeto VEMMI (no componente) en la pantalla del terminal VEMMI. Si una instrucción VEMMI_Cambiar_ubicación_objeto se refiere a un objeto cerrado VEMMI, no habrá efectos visibles. Este elemento de servicio no cambiará el parámetro de estado activo/inactivo de un objeto.

El elemento de servicio VEMMI_Cambiar_ubicación_objeto no se aplicará nunca a una barra de aplicaciones. Véase el Cuadro 18.

CUADRO 18/T.107

Parámetro

Parámetros	Descripción
Número de identificación de objeto	
XPos	Nueva posición horizontal
YPos	Nueva posición vertical

5.2.17 VEMMI_Cargar_tabla_col

Este elemento de servicio lo utilizará el anfitrión para definir los colores de la tabla de colores.

Los colores se definen consecutivamente, comenzando por el índice «entrada col». El terminal enviará un mensaje de error al anfitrión cuando los colores solicitados no se pueden definir, por ejemplo, porque el dispositivo soporta menos colores que los pedidos por el anfitrión. Véase el Cuadro 19.

CUADRO 19/T.107

Parámetros

Parámetros	Descripción
Lista col	Forma de índice de color por la que los colores se definen en orden ascendente
Lista col RGB	La lista consiste en tripletes que dan el componente rojo, verde y azul de un color. Cada valor de componente está dentro de la gama 0 .. 63. El valor 0 significa componente sin intensidad, el valor 63 significa intensidad máxima

NOTA – Si el terminal no soporta este servicio, lo comunicará a la aplicación en uso mediante VEMMI_Error «servicio no soportado».

5.2.18 VEMMI_Restablecer_tabla_col

Este elemento de servicio lo utilizará el anfitrión para restablecer todos los colores definidos previamente. Esta instrucción no tiene parámetros. Sólo los colores por defecto (índices 0 .. 15) se mantienen definidos. Si los colores por defecto fueron redefinidos con VEMMI_Cargar_tabla_col, esta instrucción fija los colores por defecto a sus valores predefinidos. Véase el Cuadro 20.

CUADRO 20/T.107

Parámetros

Parámetros	Descripción
Ninguno	

Los colores por defecto se ofrecen en el Cuadro 21.

CUADRO 21/T.107

Parámetros

Índice de color	Rojo	Verde	Azul	Color
0	0	0	0	Negro
1	0	0	42	Azul
2	0	42	0	Verde
3	0	42	42	Cian
4	42	0	0	Rojo
5	42	0	42	Magenta
6	42	21	0	Marrón
7	42	42	42	Blanco
8	21	21	21	Gris
9	21	21	63	Azul claro
10	21	63	21	Verde claro
11	21	63	63	Cian claro
12	63	21	21	Rojo claro
13	63	21	63	Magenta claro
14	63	63	21	Amarillo
15	63	63	63	Blanco brillante

NOTA – Si el terminal no soporta este servicio, lo comunicará a la aplicación en uso mediante VEMMI_Error «servicio no soportado».

5.2.19 VEMMI_Abrir_aplicación

Este elemento de servicio lo utilizará el anfitrión para cargar un conjunto de elementos y definir el nombre y los atributos de la aplicación en curso. Puede proporcionarse soporte adicional utilizando nombres de directorio simbólicos, que se describen en 4.12. Los conjuntos de objetos almacenados localmente se identifican mediante el identificador de aplicación. Se utiliza una indicación de hora para asociar información sobre la versión con los objetos de esta aplicación. Véase el Cuadro 22.

CUADRO 22/T.107

Parámetros

Parámetros	Descripción
Id. de aplicación	Cadena con el nombre de la aplicación
Datos adec. de aplicación	Datos adicionales de identificación de aplicación
Indicación de hora	Hora en segundos desde 1.1.1970, Hora Media de Greenwich (GMT)

El parámetro ApplId contendrá el nombre de la aplicación. En el parámetro ApplAddData, que no es obligatorio, se puede suministrar información adicional, como nombre de la empresa u organización a la que pertenece la aplicación, o el autor de la aplicación. El terminal no necesita analizar el contenido de ApplId, ya que sólo se utiliza para identificar la aplicación solicitada.

Los proveedores de aplicaciones deben procurar elegir identificadores de aplicación únicos, a fin de evitar la sobrescritura de otras aplicaciones.

5.2.20 VEMMI_Borrar_objeto_antiguado

Este elemento de servicio lo utilizará el anfitrión para pedir la supresión, en la memoria del terminal, de objetos locales anticuados de la aplicación en curso. Véase el Cuadro 23.

Si el anfitrión envía una instrucción VEMMI_Borrar_objeto_antiguado con los requisitos necesarios de índice de tiempo, el terminal reaccionará así:

- si un objeto es referenciado con un valor de índice de tiempo mayor que el valor de índice de tiempo del objeto almacenado localmente, se suprimirá ese objeto del almacenamiento local. Por consiguiente, la aplicación anfitriona puede estar segura de que los objetos anticuados ya no se utilizan.

CUADRO 23/T.107

Parámetros

Parámetros	Descripción
Lista actualizada	Este parámetro contiene grupos, cada uno de dos elementos, una indicación de hora y un conjunto de objetos

Para actualizar parcialmente una aplicación, esta instrucción suele enviarse después de VEMMI_Abrir_aplicación.

5.2.21 VEMMI_Almacenar_objetos

Este elemento de servicio lo utilizará el anfitrión para pedir al terminal que almacene los objetos, o parte de ellos, cargados en ese momento. A cada objeto se le asocia la indicación de hora de la aplicación abierta en ese momento, proporcionada por la instrucción VEMMI_Abrir_aplicación. Véase el Cuadro 24.

CUADRO 24/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	

Si el parámetro no figura, se almacenan todos los objetos cargados en ese momento.

5.2.22 VEMMI_Borrar_objetos

Este elemento de servicio lo utilizará el anfitrión para pedir la supresión, en el almacenamiento local de un terminal, de los objetos de la aplicación en curso. Para aumentar la seguridad de la implementación, el terminal VEMMI podría preguntar al usuario antes de ejecutar la supresión (por ejemplo, una vez al comienzo de la aplicación o con cada supresión). Si no hay ninguna aplicación abierta, la instrucción no tiene ningún efecto. Véase el Cuadro 25.

CUADRO 25/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	

5.2.23 VEMMI_Bloquear_usuario

Este elemento de servicio se utilizará para impedir cualquier entrada de usuario en el terminal hasta que se envíe una instrucción VEMMI_Desbloquear_usuario. Véase el Cuadro 26.

CUADRO 26/T.107

Parámetros

Parámetros	Descripción
Ninguno	

5.2.24 VEMMI_Desbloquear_usuario

Este elemento de servicio se utilizará para permitir entradas de usuario en el terminal. Véase el Cuadro 27.

CUADRO 27/T.107

Parámetros

Parámetros	Descripción
Ninguno	

5.2.25 VEMMI_Transferencia_recursos

Este elemento de servicio se utilizará para transferir ficheros de transferencia referenciados por objetos recurso VEMMI desde una aplicación VEMMI a un terminal VEMMI. Véase el Cuadro 28.

Un fichero puede ser telecargado utilizando una o más instrucciones VEMMI_Transferencia_recursos. Un parámetro de la instrucción VEMMI_Transferencia_recursos (ID de transferencia) asocia la instrucción a una determinada transferencia de recursos, lo cual permite efectuar simultáneamente varias transferencias de recursos con instrucciones pertenecientes a diferentes transferencias de recursos entrelazadas entre sí sin interferencia.

Las instrucciones VEMMI_Transferencia_recursos pertenecientes a la misma transferencia de recursos se numeran consecutivamente utilizando el parámetro número de bloque (BlockNumber).

El primer bloque de una determinada transferencia de recursos contendrá la siguiente información de encabezamiento del fichero:

- nombre de fichero;
- tamaño de fichero;
- número de instrucciones VEMMI_Transferencia_recursos utilizadas para transmitir el fichero;
- fecha de creación.

El parámetro tipo de transferencia (TransferType) de la instrucción VEMMI_Transferencia_recursos indica si la instrucción VEMMI_Transferencia_recursos de una determinada transferencia de recursos es:

- la primera instrucción;
- una instrucción intermedia; o
- si la transferencia de recursos es abortada por la aplicación VEMMI.

CUADRO 28/T.107

Parámetros

Parámetros	Descripción
Tipo de transferencia	Primera instrucción, instrucción intermedia, aborto
ID de transferencia	Identificador de transferencia
Número de bloque	Número de la instrucción dentro de una determinada transferencia de recursos
Información de fichero	Información de encabezamiento acerca del fichero
Bloques totales	Número de instrucciones de VEMMI_Transferencia_recursos utilizadas para transmitir un determinado fichero

5.3 Elementos de servicio iniciados por el terminal

5.3.1 VEMMI_Identificar_resp_cap_term

Este elemento de servicio se utilizará para responder a una petición VEMMI_Identificar_resp_cap_term. Véase el Cuadro 29.

Podrá ofrecer la información siguiente:

- identificación de la versión VEMMI;
- soporte de almacenamiento local (soportado/no soportado);
- lista de idiomas de usuarios preferidos (cadena);
- tipo de sistema (por ejemplo: IBM compatible, Macintosh, ...) sistema operativo (por ejemplo: MS-DOS, OS2, Microsoft Windows, ...), versión de sistema operativo (por ejemplo: 3.1) (cadena);
- una lista de los tipos de datos soportados (referencia al catálogo de identificación de codificación de contenido VEMMI).

CUADRO 29/T.107

Parámetros

Parámetros	Descripción
Versiones VEMMI soportadas	Este parámetro contiene la identificación de la versión VEMMI
Lista de contenido	La lista da referencias de los tipos de datos soportados en el catálogo de identificación de codificación de contenido VEMMI
Almacenamiento local	Identifica el soporte de la facilidad de almacenamiento local
Idiomas preferidos por el usuario	Idiomas preferidos por el usuario (en aplicaciones multilingües) El primer idioma de la lista es el más preferido. La lista puede contener sólo un idioma
Info de sistema	Información para soportar la correcta utilización de los objetos operativos (por ejemplo, tipo de sistema, sistema operativo, versión de sistema operativo)

5.3.2 VEMMI_Retransmisión_objeto

Este elemento de servicio se utilizará para pedir a la aplicación VEMMI que retransmita un objeto VEMMI. Se utilizará solamente después de:

- la recepción de la instrucción VEMMI_Abrir_objeto;
- la instrucción VEMMI_Abrir_objeto_bloqueante;
- una acción local «abrir objeto»,

que se refieren a un objeto que no existe en el terminal.

Hasta que se reciba el objeto solicitado, el terminal aplicará una de las reglas siguientes al proceso de las instrucciones VEMMI:

Regla A

- informar al usuario sobre posible retardo debido a una retransmisión (dependiente del terminal);
- enviar una petición VEMMI_Retransmisión_objeto;
- ejecutar todas las instrucciones y acciones locales VEMMI no referidas al objeto solicitado;
- memorizar todas las instrucciones y acciones locales VEMMI referidas al objeto solicitado.

Después de que se ha vuelto a crear y abrir el objeto solicitado, el terminal:

- reanudará la ejecución de las acciones locales posiblemente suspendidas;
- reanudará la ejecución de las instrucciones VEMMI posiblemente memorizadas recibidas en el orden de su recepción.

Regla B

- bloquear al usuario;
- informar al usuario sobre un posible retardo debido a una retransmisión (dependiente del terminal);
- enviar una petición VEMMI_Retransmisión_objeto;
- suspender la ejecución de todas las instrucciones o acciones locales VEMMI.

Después de que se ha vuelto a crear y abrir el objeto solicitado, el terminal:

- reanudará la ejecución de las acciones locales posiblemente suspendidas;
- reanudará la ejecución de las instrucciones VEMMI posiblemente memorizadas recibidas en el orden de su recepción;
- desbloqueará al usuario.

Véase el Cuadro 30.

CUADRO 30/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	

5.3.3 VEMMI_Datos_usuario

Este elemento de servicio se utilizará para enviar a la aplicación VEMMI datos de usuario correspondientes a un objeto. Véase el Cuadro 31.

CUADRO 31/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	
Datos comp.	Entradas de usuario

5.3.4 VEMMI_Resp_abrir_aplicación

Este elemento de servicio será utilizado por el terminal para responder a una instrucción VEMMI_Abrir_aplicación. Véase el Cuadro 32.

CUADRO 32/T.107

Parámetros

Parámetros	Descripción
Resultado aplic. abierta	Verdadero: La aplicación ha sido identificada y los objetos se han cargado Falso: La aplicación no ha sido identificada y no se ha cargado ningún objeto. Los parámetros usados en la instrucción VEMMI_Abrir_aplicación se utilizan para describir un nuevo entorno de aplicación

5.3.5 VEMMI_Resp_almacenar_objetos

Este elemento de servicio será utilizado por el terminal para responder a una instrucción VEMMI_Almacenar_objetos. Véase el Cuadro 33.

CUADRO 33/T.107

Parámetros

Parámetros	Descripción
Resultado de almacenamiento	Verdadero: Los objetos se han almacenado Falso: Los objetos no se han almacenado

5.3.6 VEMMI_Error

Este elemento de servicio será utilizado solamente por el terminal VEMMI para informar a la aplicación VEMMI de diferentes situaciones de error.

Una instrucción VEMMI_Error con el parámetro Tipo de error = «No hay suficiente memoria» se refiere sólo al objeto especificado. Pueden almacenarse en el terminal otros objetos creados después de éste. Véase el Cuadro 34.

CUADRO 34/T.107

Parámetros

Parámetros	Descripción
Número de identificación de objeto	
Tipo de error	
Número de identificación de comp.	
Código de instrucción de error	Identificación de la instrucción que ha causado el error

Se definen los siguientes tipos de error:

- Error general:
Se ha producido un error general.
- Instrucción VEMMI desconocida:
El código de instrucción VEMMI no existe.
- Instrucción VEMMI errónea:
La instrucción VEMMI recibida no tiene todos los parámetros obligatorios o éstos tienen valores erróneos.
- Error de sintaxis de objeto:
La descripción de un objeto VEMMI no es correcta en una creación o modificación de componente.
- Instrucción VEMMI inesperada:
La instrucción VEMMI recibida es correcta pero se produce en un momento erróneo.
- No hay suficiente memoria:
El terminal no tiene suficiente memoria para almacenar los datos correspondientes a una creación de objeto o modificación de componente.
- Servicio no soportado:
El servicio solicitado no es soportado.
- Objeto no soportado.
- Tipo de contenido de datos no soportado.
- Índice de color no válido.
- Fichero no encontrado.
- Conversión a mapa de bits fracasada.
- Indicación de destrucción de objeto (un objeto cerrado fue destruido por el terminal).
- No hay suficiente espacio de almacenamiento local permanente.

5.3.7 VEMMI_Aborto_transferencia_recurso

Este elemento de servicio será utilizado por el terminal VEMMI para abortar un recurso. Véase el Cuadro 35.

CUADRO 35/T.107

Parámetros

Parámetros	Descripción
ID de transferencia	Identificador de transferencia

6 Presentación de objeto VEMMI

Los objetos VEMMI ofrecen una selección básica de los elementos de diálogo que necesitan las aplicaciones VEMMI.

En esta Recomendación se definen los objetos VEMMI siguientes:

- barra de aplicaciones;
- barra de botones;
- menú superpuesto;
- casilla de diálogo;

- casilla de mensajes;
- objeto operativo;
- objeto recurso mapa de bits;
- objeto recurso videotex;
- objeto recurso texto;
- objeto recurso tipo de caracteres;
- objeto metacódigo.

Una aplicación VEMMI puede diseñarse utilizando cualquiera de los objetos definidos. Ningún objeto o componente VEMMI determinado es obligatorio dentro de una aplicación VEMMI. Todos los objetos VEMMI pueden ser múltiples dentro de una aplicación VEMMI, salvo la barra de aplicaciones. Dentro de un objeto, todos los componentes pueden ser múltiples, salvo algunas restricciones aplicables a determinados componentes (véase la cláusula 7).

6.1 Barra de aplicaciones

La barra de aplicaciones permite al usuario seleccionar entre las diferentes partes de aplicación y partes de subaplicación VEMMI ofrecidas por la aplicación VEMMI seleccionada. Cuando se utiliza, la barra de aplicaciones es única y está situada en la parte superior (barra horizontal) o en el lado izquierdo (barra vertical) de la DDA.

6.1.1 Composición

La barra de aplicaciones se subdivide en tres grupos lógicos diferentes de componentes de selección por menú. Estos grupos difieren en su comportamiento y funcionalidades. Los tres grupos diferentes se denominan:

- barra (bar);
- menú desplegable (pull-down menu);
- menú en cascada (cascading menu).

La barra es una lista horizontal o vertical de componentes de selección por menú que representan las diferentes partes de la aplicación VEMMI.

Los menús desplegables son listas verticales de componentes de opción de menú asociados con el mismo componente de opción de menú de la barra. Los menús desplegables representan las diferentes partes de subaplicaciones de la aplicación VEMMI.

Los menús en cascada son listas verticales de componentes de opción de menú que están asociadas con el mismo componente de opción de menú del menú desplegable. Los menús en cascada representan las diferentes partes de subaplicación de la aplicación VEMMI.

6.2 Barra de botones

La barra de botones permite seleccionar entre un conjunto de alternativas, en un momento dado, durante la ejecución de la aplicación VEMMI. Cada opción está representada por un botón. La barra de botones puede estar situada en cualquier parte de la DDA y puede ser horizontal o vertical.

6.2.1 Composición

La barra de botones está formada por una serie de componentes, denominados botones.

6.3 Menú superpuesto

El menú superpuesto ofrece opciones y subopciones apropiadas para un elemento VEMMI dado en su contexto vigente. El menú superpuesto puede estar ubicado en cualquier parte de la DDA.

6.3.1 Composición

El menú superpuesto se subdivide en dos diferentes grupos lógicos de componentes de selección por menú. Estos grupos difieren en su comportamiento y funcionalidades, pero no en su descripción. Los dos grupos diferentes se denominan:

- menú superpuesto primario;
- menú en cascada.

El menú superpuesto primario es una lista vertical de componentes de opción de menú que ofrecen opciones apropiadas para un elemento VEMMI dado en su contexto actual.

El menú en cascada es una lista vertical de componentes de opción de menú asociados con el mismo componente de opción de menú superpuesto primario. El menú en cascada ofrece subopciones apropiadas para un elemento VEMMI dado en su contexto vigente.

6.4 Casilla de diálogo

La casilla de diálogo es el objeto donde se produce la principal interacción entre el usuario y la aplicación VEMMI. Para permitir esta interacción, se define un conjunto de componentes, que se pueden clasificar como componentes de presentación o de diálogo.

Los componentes de presentación son inaccesibles; su finalidad es sólo presentar los diferentes componentes de diálogo de manera coherente y atractiva.

Los componentes de diálogo permiten la interacción entre el usuario y la aplicación VEMMI.

Se definen cinco componentes de presentación:

- el separador;
- el marco;
- la zona de presentación de texto;
- el componente de texto;
- la zona de salida gráfica.

Se definen nueve componentes de diálogo:

- el botón pulsado;
- el campo de entrada de texto;
- la casilla de marca;
- el botón radio;
- la casilla de lista;
- la casilla de combinación;
- el deslizador;
- la zona sensible;
- el texto sensible.

6.4.1 Composición

6.4.1.1 Componente separador

Un separador es una línea de trazo continuo horizontal o vertical. Su objetivo es separar visualmente diferentes componentes de diálogo dentro de la caja de diálogo.

6.4.1.2 Componente marco

Un marco es un elemento de presentación para separar visualmente una zona determinada de la caja de diálogo y sus diferentes componentes.

6.4.1.3 Componente zona de presentación de texto

La zona de presentación de texto es una zona rectangular en la cual se visualizan datos de texto. Los objetivos de este componente son:

- presentar información de texto;
- titular o etiquetar componentes de diálogo;
- presentar resultados de la ejecución de la aplicación VEMMI.

6.4.1.4 Componente texto

La finalidad de este componente es separar datos de texto grandes en unidades (componentes de texto) y definir la información estructural necesaria (concatenación de los componentes de texto) para la visualización en una zona de texto.

6.4.1.5 Componente área de salida gráfica

La finalidad de este componente es presentar datos gráficos al usuario. Se soportan varios formatos de codificación de datos gráficos.

6.4.1.6 Componente texto sensible

La finalidad de este componente es definir las operaciones de activación y validación de las cadenas de texto sensible.

6.4.1.7 Componente botón pulsado

La finalidad del botón es activar una acción local que se realizará inmediatamente.

6.4.1.8 Componente campo de entrada de texto

La finalidad del campo de entrada de texto es recopilar datos de texto, introducidos por el usuario. Es una zona rectangular compuesta de una etiqueta de texto asociada con una zona de entrada.

6.4.1.9 Componente casilla de marca

La finalidad de la casilla de marca es introducir y visualizar una opción de usuario independiente. Una casilla de marca mantiene el valor marcado o no marcado independientemente de cualesquiera otras casillas de marca.

6.4.1.10 Componente botón radio

La finalidad del botón radio es introducir y visualizar una opción de usuario. El botón radio permite una sola opción entre varias posibilidades ofrecidas en un grupo de botones radio. La marca de un botón radio hace que no se marquen los otros botones radio pertenecientes al mismo grupo de botones radio.

6.4.1.11 Componente casilla de lista

La finalidad de la casilla de lista es ofrecer una opción o múltiples opciones entre una lista de ítems de texto. La lista en general no es totalmente visible al usuario, por lo que se ofrecen diferentes controles para desplazar la lista en sentido ascendente y descendente.

6.4.1.12 Componente casilla de combinación

La finalidad de la casilla de combinación es combinar la funcionalidad de una casilla de lista de una sola opción con la funcionalidad de un campo de entrada de texto. Contiene una lista de ítems de texto que el usuario puede desplazar verticalmente a través de todo el campo de entrada de texto. Un parámetro de la casilla de combinación especifica si el contenido del campo de entrada de texto puede ser editado o no. Si el contenido del campo de introducción de texto puede ser editado, el usuario puede mecanografiar el texto directamente en el campo de entrada de texto.

Una variación de la casilla de combinación es una casilla de combinación de exclusión (Drop Down Combination Box), que se compone de una casilla de combinación y un botón pulsado. Sólo se visualizan el campo de entrada de texto y el botón pulsado hasta que el usuario selecciona el botón pulsado asociado. La validación del botón pulsado genera la visualización de la casilla de lista asociada.

6.4.1.13 Componente deslizador

El deslizador permite seleccionar un valor análogo desplazando un marcador ajustable sobre una barra de deslizamiento entre un valor máximo y uno mínimo. Los intervalos los fija la aplicación.

6.4.1.14 Componente zona sensible

La finalidad de la zona sensible de la casilla de diálogo es ofrecer una zona de selección asociada a una zona de salida.

6.5 Objeto operativo

Con este objeto, una aplicación referencia un programa que se vinculará a la aplicación VEMMI. Este tipo de objeto proporciona un método para ampliar las capacidades de una aplicación durante el tiempo de ejecución.

6.6 Objeto recurso mapa de bits

Un objeto mapa de bits contiene la propia definición de mapa de bits o sólo una referencia a un fichero que contiene la definición de mapa de bits.

6.7 Objeto recurso videotex

Un objeto videotex contiene el contenido videotex propiamente dicho o sólo una referencia a un fichero con el contenido videotex.

6.8 Objeto recurso texto

Este objeto define el contenido de texto como un recurso que puede ser referenciado mediante (número de identificación de texto) (TIN). Contiene el propio contenido de texto o una referencia a un fichero con el contenido de texto.

6.9 Objeto recurso tipo de caracteres

Este objeto combina un conjunto de atributos de texto sobre el recurso tipo de caracteres, que puede ser referenciado mediante FIN.

6.10 Objeto metacódigo

El objeto metacódigo contiene instrucciones VEMMI. Este objeto proporciona un medio fácil para evitar pasos de diálogo innecesarios con la aplicación anfitriona.

6.11 Casilla de mensajes

La finalidad de la casilla de mensajes es presentar información, no solicitada por un usuario pero enviada por la aplicación VEMMI, en respuesta a un evento inesperado, o cuando pudiera ocurrir algo indeseable.

7 Descripción funcional

7.1 Reglas generales para el comportamiento de los elementos

7.1.1 Interacción de usuario

El usuario tendrá la posibilidad de acceder a los diferentes elementos VEMMI y activarlos. El terminal permitirá también al usuario cambiar los valores de los elementos VEMMI y validar estas entradas.

7.1.2 Acciones locales e informes

El informe de entrada de usuario del terminal al servidor será generado por una instrucción de informe definida en una acción local por la aplicación VEMMI. Al ejecutar una instrucción de informe, el terminal enviará una instrucción VEMMI_Datos_usuario a la aplicación VEMMI. Los eventos generadores que producen la ejecución de acciones locales son:

- activación de un componente;
- validación de un componente.

Las instrucciones de informe disponibles son:

- comunicar OIN + CIN;
- comunicar los valores actuales del componente;
- comunicar los valores de todos los componentes de este objeto;
- comunicar todos los valores de todos los componentes de los objetos progenitores que han cambiado (también mediante una VEMMI_Modificar_componente) desde el último informe.

Los Cuadros 36 y 37 muestran los posibles eventos generadores y los posibles informes para cada componente. La estructura global de las acciones locales se define en 4.8.

CUADRO 36/T.107

Eventos generadores

Componente	Activación	Validación
Opción de menú barra	✓	✓
Opción de menú desplegable	✓	✓
Opción de menú en cascada	✓	✓
Opción de menú superpuesto	✓	✓
Separador de opción de menú		
Botón	✓	✓
Separador		
Marco		
Zona de presentación de texto		
Componente texto		
Botón pulsado	✓	✓
Campo de entrada de texto	✓	✓
Casilla de marca	✓	✓
Botón radio	✓	✓
Casilla de lista	✓	✓
Casilla de combinación	✓	✓
Deslizador	✓	✓
Zona sensible	✓	✓
Texto sensible		
Zona de salida gráfica		

7.1.3 Relación entre objetos y componentes

El estado cerrado de un objeto invalida el estado abierto de un componente. Si se cierra un objeto, se suprime por completo de la pantalla, incluidos todos sus componentes, independientemente de si éstos están abiertos o cerrados. El estado abierto de un objeto no invalida el estado cerrado de un componente. Si se abre un objeto, no se visualizan los componentes que están en estado cerrado.

El parámetro de estado inaccesible de un objeto invalida el parámetro de estado accesible de un componente. Si un objeto se pone en inaccesible, el objeto entero, incluidos todos sus componentes, se tornan inaccesibles, sin tener en cuenta si los componentes son accesibles o no. El parámetro estado accesible de un objeto no invalida el parámetro estado inaccesible de un componente. Si un objeto se pone en accesible, el usuario sólo podrá interactuar con los componentes que están accesibles.

CUADRO 37/T.107

Informes

Componente	CIN	Valor de todos los componentes	Valor (Nota)
Opción de menú barra	✓		
Opción de menú desplegable	✓		
Opción de menú en cascada	✓		
Opción de menú superpuesto	✓		
Separador de selección por menú			
Botón	✓		
Opción de menú	✓		
Separador			
Marco			
Zona de presentación de texto			
Componente texto			
Botón pulsado	✓	✓	
Campo de entrada de texto	✓	✓	Cadena
Casilla de marca	✓	✓	Booleano
Botón radio	✓	✓	Booleano
Casilla de lista	✓	✓	Lista de cadena
Casilla de combinación	✓	✓	Cadena, entero
Deslizador	✓	✓	Entero, booleano
Zona sensible	✓	✓	
Texto sensible			
Zona de salida gráfica			
NOTA – Las entradas en esta columna especifican el tipo de datos de los valores comunicados.			

7.1.4 Abrir/cerrar objetos audio, vídeo, recurso y metacódigo

Si se abre un objeto sonido, se ejecutará el sonido. Al terminar la salida sonido, el objeto se considerará cerrado.

La misma regla se aplica a los objetos vídeo. Si se abre el objeto, comienza el proceso de visualización de vídeo, y cuando termina, se cierra el objeto.

La operación abrir de un recurso sólo verifica si está presente en el terminal.

Un objeto metacódigo es procesado mediante la operación abrir. Al terminar el procesamiento, se considera cerrado.

7.1.5 Operación maximizar

NOTA – La operación que se describe en la subcláusula siguiente no es obligatoria para un terminal VEMMI. Si un terminal VEMMI no realiza esta operación, ignorará los atributos y operaciones correspondientes.

Un objeto puede definirse como maximizable. En este caso, es visualizado con un botón de maximizar en su fila de título, y al validar ese botón, se visualiza agrandado. El objeto agrandado se completa con un botón de restablecimiento, y al validar ese botón, el objeto es nuevamente visualizado en su tamaño inicial. Estas operaciones dependen del terminal y el UI, especialmente la de maximización. La aplicación anfitriona sólo puede marcar un objeto como maximizable.

Todos los elementos y componentes de objeto, salvo la zona de presentación de texto, se amplían conforme a las especificaciones del terminal. El componente zona de presentación de texto de una casilla de texto tiene un comportamiento específico para que un objeto agrandado pueda añadir localmente más contenido de texto en su zona de visualización. Si el componente campo de salida de texto contiene todo el texto en su tamaño inicial, la operación maximizar depende del terminal. Si el texto es visualizado sólo parcialmente en su posición inicial, el terminal añadirá texto adicional al objeto maximizado. El objeto agrandado puede visualizar el texto con un formato nuevo, que puede ser fijado por el anfitrión.

Si se cierra el objeto con su tamaño agrandado y se vuelve a abrir más adelante, se visualiza con su tamaño inicial (no ampliado). Si se desactiva un objeto maximizado, éste mantiene su forma.

Pueden ofrecerse al usuario operaciones adicionales locales sobre objetos, que no son definibles por el anfitrión y que dependen del terminal. Éstas pueden ser: minimizar, desplazar, iconizar, aumento y disminución unidireccionales, etc.

7.1.6 Convenios de notación

En las próximas subcláusulas se aplican los convenios de notación siguientes:

- 1) En las subcláusulas siguientes de esta cláusula, cada descripción de objeto y de componente contiene una lista de atributos. Esas listas contienen los atributos necesarios para describir las propiedades de los elementos. La sintaxis precisa de la codificación de los elementos se define en la cláusula 9.
- 2) Los números de identificación de los objetos recurso: BIN (número de identificación de mapa de bits), FIN (número de identificación de tipo de caracteres), TIN (número de identificación de texto) y VIN (número de identificación videotex) son casos especiales de OIN y se utilizan en el mismo nivel, por ejemplo, como parámetros de las instrucciones VEMMI. Por el contrario, CIN siempre designa un componente.

Los convenios de denominación de los atributos booleanos en la descripción de elemento VEMMI siguen las reglas siguientes:

- Si el atributo está presente en la codificación, su valor booleano corresponde al valor expresado por su nombre.

Ejemplo

El valor por defecto del estado «abierto» es verdadero. Si el atributo «cerrado» aparece en la codificación de un elemento VEMMI, el elemento estará en el estado «cerrado». Si el atributo «cerrado» no aparece en la codificación de un elemento VEMMI, el elemento estará en su estado por defecto: el estado «abierto».

7.1.7 Mnemónica

Cuando en un componente se utilizan datos de texto, el texto puede contener el carácter «&», que señala al carácter siguiente como clave mnemónica. La clave mnemónica puede emplearse para validar o activar el componente. Para presentar el carácter «&» en un texto se duplicará (por ejemplo, contenido de texto =«&gatos && perros»; texto presentado al usuario «gatos & perros»).

7.2 Formato de texto

7.2.1 Texto VEMMI de alta calidad

La implementación de atributos de texto (cambio de tipo de caracteres, tamaño del texto, etc., dentro de una cadena de texto), y la definición de zonas sensibles de texto (texto sensible o *hot spot*, en inglés) se realiza mediante atributos in-text (dentro del texto). Esos atributos son aplicables a los elementos siguientes:

- componentes zona de texto de una casilla de diálogo;
 - etiquetas.
- 1) *Atributos de texto*

Puede cambiarse el tipo de caracteres atribuido dentro de una operación de salida de texto. El tipo de caracteres es referenciado mediante FIN. El nuevo tipo de caracteres seleccionado se aplica a todo el texto subsiguiente hasta el próximo atributo in-text que cambie el tipo de caracteres, o hasta el fin del componente. El tipo de caracteres inicial atribuido al primer texto del componente se define como un atributo adicional del componente.

El ejemplo siguiente ilustra cómo se emplean los atributos in-text. Los corchetes angulares, <...>, indican atributos de los componentes con su parámetro.

Ejemplo

.....

<text1> = «nota de servicio para el cliente:(CR,LF,CR,LF)»

<use FIN 5>

<text2> = «negativos tamaño 110»

<use FIN 6>

<text3> = «no deben cortarse, sino devolverlos en forma de tira para su nueva reimpresión»

<use FIN 5>

<text4> = «si se cortan, no podremos sacar copias de ellos»

<use FIN 7>

<text5> = «(CR,LF,CR,LF)Emitida por A.P.L.»

.....

FIN 5 se define como ROMAN, altura 12.

FIN 6 se define como ROMAN, altura 12, negrita, subrayado.

FIN 7 se define como SWISS, altura 8.

Esta secuencia de texto con los atributos in-text podría quedar así:

Nota de servicio para el cliente:

Los negativos de tamaño 110 *no deben cortarse, sino devolverlos en forma de tiras para su reimpresión*. Si se cortan, no podremos sacar copias de ellos.

Emitida por A.P.L.

2) Atributos in-text para definir el texto sensible

Cada cadena de texto definida como sensible se refiere a un componente texto sensible. Este componente contiene dos atributos que definen las operaciones de activación y validación que habrá que efectuar cuando el usuario interactúa con texto sensible.

Los dos grupos de atributos pueden mezclarse en un componente de texto, de manera que la aplicación puede aplicar atributos de texto, por ejemplo, color, a las cadenas sensibles. Esas cadenas podrían aparecer en color verde, y quizás realizadas además, para distinguirlas fácilmente del texto, que podría ser negro sobre blanco. Con estos atributos, la aplicación puede ofrecer al usuario una salida de tipo «hipertexto» con VEMMI.

7.2.2 Etiquetas y títulos de texto

El contenido de texto de etiquetas y títulos se recorta en los límites de sus contenedores. Esto se aplicará a los atributos siguientes:

- todos los atributos de título;
- todos los atributos de etiqueta;
- atributo de contenido de texto de los componentes opción de menú;
- atributo de texto de lista de los componentes casilla de lista y casilla de combinación.

7.3 Barra de aplicaciones

La barra de aplicaciones se subdivide en tres diferentes grupos lógicos de componentes de selección por menú. Estos grupos difieren en su comportamiento y funcionalidad. Los tres grupos diferentes se denominan.

- barra;
- menú desplegable;
- menú en cascada.

NOTA 1 – La representación vertical de la barra de aplicaciones es una característica opcional de un terminal VEMMI. Si el atributo vertical está presente y fijado a verdadero en la codificación de la barra de aplicaciones, el terminal puede ignorarlo y presentar la barra de aplicaciones horizontalmente. No se enviará ningún mensaje de error al anfitrión.

NOTA 2 – Todos los atributos de posicionamiento y dimensionamiento de la barra de aplicaciones y de todos sus componentes de opción de menú son opcionales. Si los atributos de posicionamiento y dimensionamiento se definen en la codificación de la barra de aplicaciones, el terminal puede ignorarlos y presentar la barra de aplicaciones comenzando por la esquina superior izquierda de la DDA, y dimensionarla conforme a los requisitos de espacio del contenido de texto. No se enviará ningún mensaje de error al anfitrión.

La barra es una lista horizontal o vertical de opción de menú. Los menús desplegables son listas verticales de componentes de opción de menú que están asociadas con el mismo componente de opción de menú de la barra. Los menús en cascada son listas verticales de componentes de opción de menú asociadas con el mismo componente opción de menú del menú desplegable.

Todos los componentes de opción de menú de la barra de aplicaciones tendrán contenido de texto únicamente.

Un componente separador de opción de menú puede utilizarse para separar visualmente componentes opción de menú en un grupo lógico. Se define fijando el atributo «separador» con la definición de una opción de menú ordinaria.

La estructura del objeto barra de aplicaciones viene dada por el orden descendente canónico de sus componentes (trayecto de izquierda a derecha, con prioridad a la profundidad).

En el siguiente ejemplo de barra de aplicaciones, el orden descendente canónico de los componentes es:

(AB(FG(IJK)H)(CDE(L)))

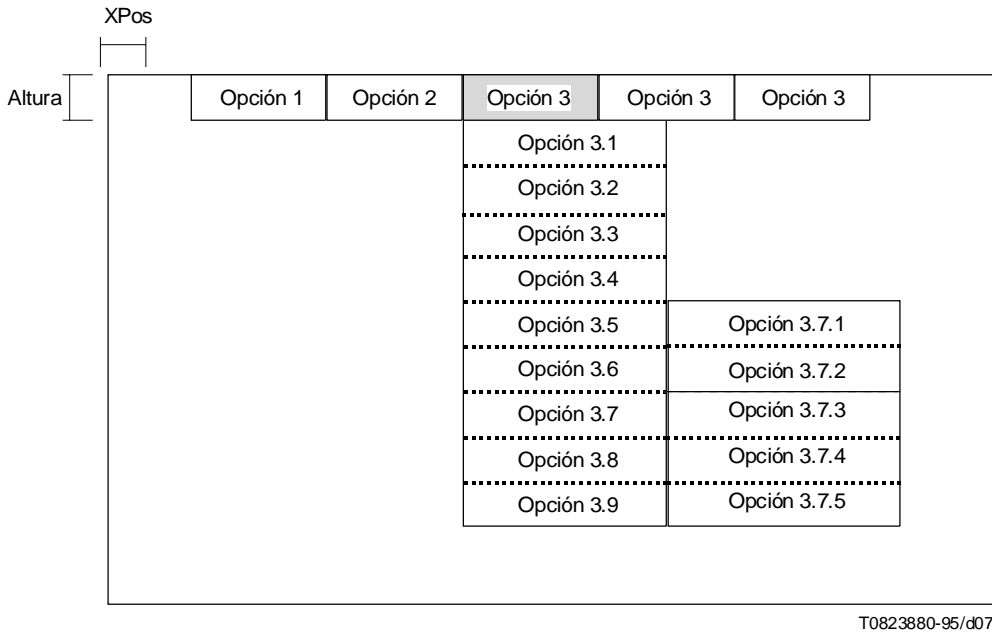
A	B	C	D	E
	F			L
	G	I		
	H	J		
		K		

NOTA 3 – Con los paréntesis del ejemplo anterior sólo se pretende mostrar los diferentes niveles de encapsulado de los diferentes grupos de objetos para el trayecto de descripción.

La descripción de la estructura del objeto es por tanto:

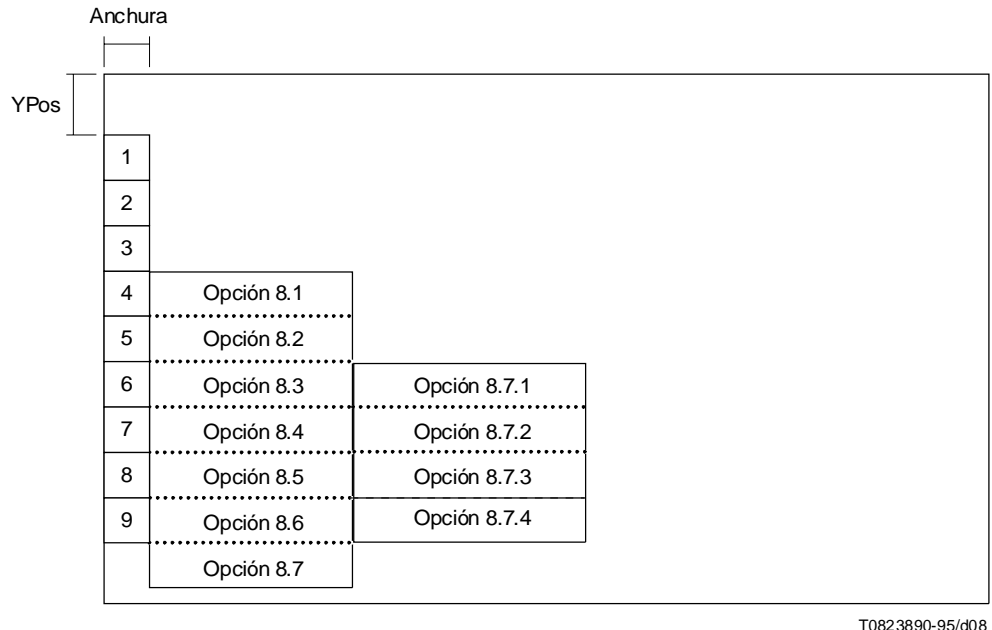
- Objeto: Barra de aplicaciones.
- Componente: Opción de menú barra A.
- Componente: Opción de menú barra B.
- Componente: Opción de menú desplegable F.
- Componente: Opción de menú desplegable G.
- Componente: Opción de menú en cascada I.
- Componente: Opción de menú en cascada J.
- Componente: Opción de menú en cascada K.
- Componente: Opción de menú desplegable H.
- Componente: Opción de menú barra C.
- Componente: Opción de menú barra D.
- Componente: Opción de menú barra E.
- Componente: Opción de menú desplegable L.

Aspecto visual general (véanse las Figuras 7 y 8)



T0823880-95/d07

FIGURA 7/T.107



T0823890-95/d08

FIGURA 8/T.107

Atributos

- **Horizontal/Vertical:** La barra de aplicaciones se presentará vertical (véase la Nota 1).
- **XPos:** Este atributo lleva la posición horizontal del elemento en la NDC. Sólo es aplicable a barras de aplicaciones horizontales (véase la Nota 2).
- **YPos:** Este atributo lleva la posición vertical del elemento en la NDC. Sólo es aplicable a barras de aplicaciones verticales (véase la Nota 2).
- **Height:** Este atributo lleva la altura de la barra. Sólo es aplicable a barras de aplicaciones horizontales (véase la Nota 2).
- **Width:** Este atributo lleva la anchura de la barra. Sólo es aplicable a barras de aplicaciones verticales (véase la Nota 2).

- FirstActive: Este atributo lleva el CIN de la opción de menú que está activa por defecto la primera vez que se abre el objeto.
- Closed: El elemento estará en el estado cerrado.
- NotAccessible: El objeto no es accesible.

7.3.1 Composición

7.3.1.1 Componentes de opción de menú de la barra

Descripción

La barra es una lista consecutiva de opciones por menú colocadas juntas horizontalmente, o verticalmente.

Representación horizontal:

- la barra de aplicaciones se presentará en la parte superior de la DDA;
- el número de filas asignadas a cada opción de menú será el mismo;

Representación vertical:

- la barra de aplicaciones se presentará en la parte más a la izquierda de la DDA;
- el número de columnas, asignadas para cada opción de menú, será el mismo.

Comportamiento

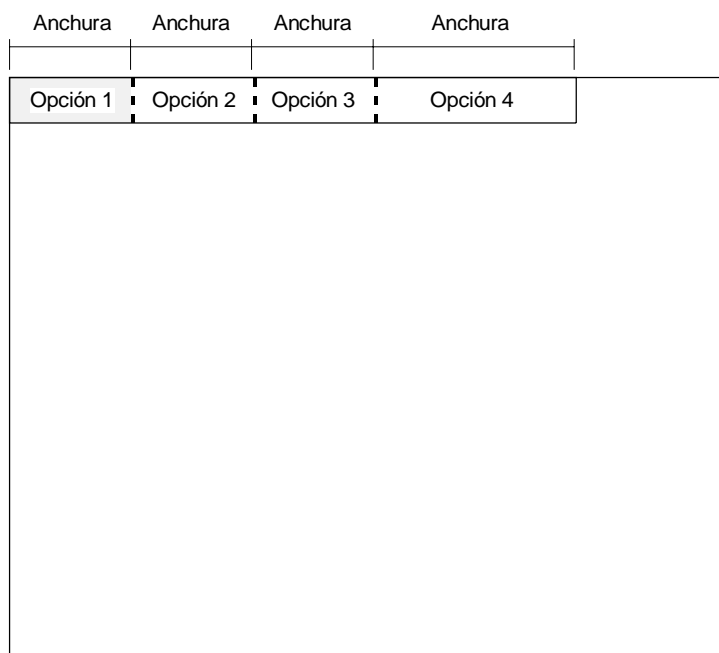
Cuando la barra de aplicaciones está activa, el terminal realzará una opción, y ofrecerá al usuario facilidades de cambio y validación. Para validar la opción de menú, puede usarse una clave mnemónica (véase 7.1.7).

Si se valida una opción de menú de la barra, se ejecutará la acción local asociada y luego se abrirá el menú desplegable correspondiente (la opción de menú se mantiene realzada).

Funcionalidad interactiva

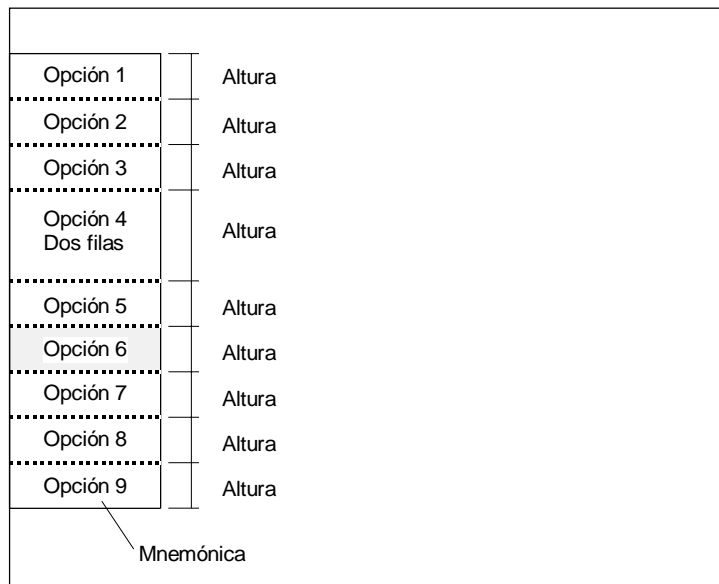
- cambio;
- validación.

Aspecto visual (véanse las Figuras 9 y 10)



T0815510-94/d09

FIGURA 9/T.107
Barra de menú horizontal



T0815520-94/d10

FIGURA 10/T.107
Barra de menú vertical

Atributos

- **CIN:** Este atributo lleva el número de identificación de componente.
- **Height:** Este atributo lleva la altura del componente. Sólo es aplicable a barras verticales (véase 7.3, Nota 2).
- **Width:** Este atributo lleva la anchura del componente. Sólo es aplicable a barras horizontales (véase 7.3, Nota 2).
- **NotAccessible:** El elemento no está accesible.
- **Text:** Este atributo lleva el contenido de texto del componente.
- **LocActAct:** Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- **LocActVal:** Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.

7.3.1.2 Componentes de opción de menú del menú desplegable

Descripción

El menú desplegable es una lista consecutiva de componentes opciones de menú asociados con la misma opción de menú de la barra y presentados verticalmente en varias filas.

Se asignará el mismo número de columnas a cada opción de menú.

El menú desplegable se colocará próximo a la opción de menú asociada de la barra.

Se puede utilizar un separador en un menú desplegable para separar visualmente diferentes opciones de menú.

Comportamiento

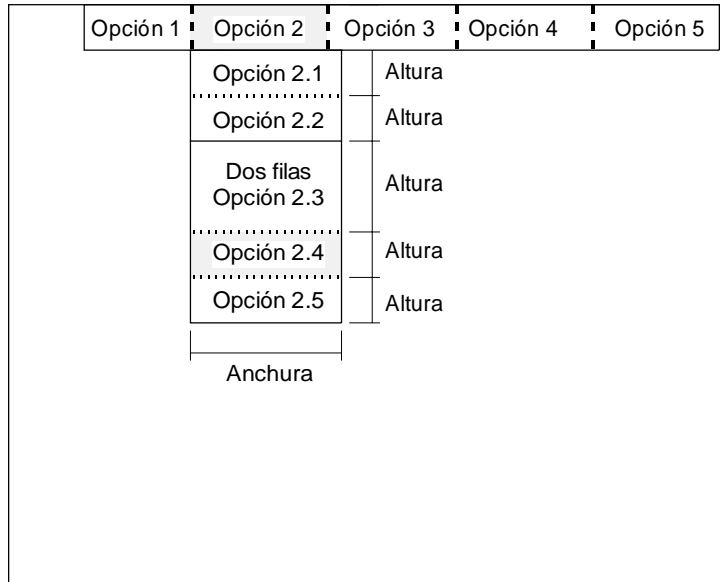
Cuando el menú desplegable está activo, el terminal realizará una opción de menú y ofrecerá al usuario facilidades de cambio y validación. Para validar la opción de menú, puede usarse una clave mnemónica (véase 7.1.7).

Si se valida una opción de menú, se ejecutará la acción local asociada y se abrirá el menú en cascada correspondiente (la opción de menú se mantiene realizada). Si después de una validación de usuario no hay un menú en cascada asociado, se ejecutará la acción local asociada y se cerrará el menú desplegable.

Funcionalidad interactiva

- cambio;
- validación.

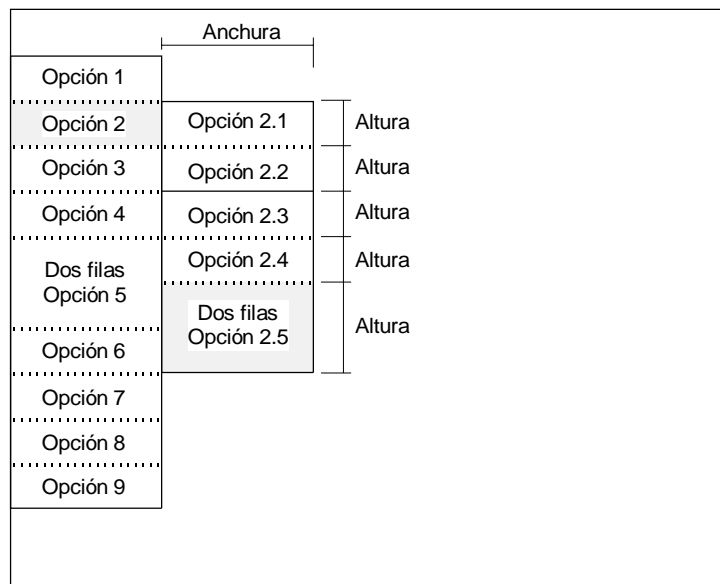
Aspecto visual (véanse las Figuras 11 y 12)



T0824320-95/d11

FIGURA 11/T.107

Menú desplegable



T0824330-95/d12

FIGURA 12/T.107

Menú desplegable

Atributos

- CIN: Este atributo lleva el número de identificación de componente.
- Height: Este atributo lleva la altura del componente (véase 7.3, Nota 2).
- Width: Este atributo lleva la anchura del componente. Será igual a todas las selecciones por menú del menú desplegable (véase 7.3, Nota 2).
- NotAccessible: El elemento no estará accesible.
- Text: Este atributo lleva el contenido de texto del componente.
- Separated (Separado) Se trazará un separador entre los elementos.
- LocActAct: Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- LocActVal: Este atributo lleva el código de la acción local asociada con el componente y generada por su validación.

7.3.1.3 Componentes de opción de menú del menú en cascada

Los componentes de opción de menú del menú en cascada son comunes al menú desplegable y al menú superpuesto.

Descripción

El menú en cascada es una lista consecutiva de opción de menú asociadas con la misma opción de menú del menú desplegable o del menú superpuesto y presentada verticalmente en varias filas.

Se asignará el mismo número de columnas a cada opción de menú.

El menú en cascada se colocará próximo al menú desplegable o menú superpuesto asociado.

Puede utilizarse un separador en un menú en cascada para separar visualmente diferentes opciones de menú.

La Figura 13 muestra la representación recomendada de un menú en cascada. Si el terminal no puede presentar un menú en cascada como el de la Figura 13 podrá, como alternativa, presentar las opciones de menú del menú en cascada en el menú desplegable, pero deberá asociar visualmente las opciones de menú del menú en cascada con las opciones correspondientes del menú desplegable. El terminal pondrá en inaccesible la opción de menú del menú desplegable con la que está asociada el menú en cascada y la presentará como una especie de título de las opciones de menú del menú en cascada (véase la Figura 14).

Comportamiento

Cuando el menú en cascada está activo, el terminal realzará una opción de menú, y ofrecerá al usuario facilidades de cambio y validación. Para validar la opción de menú puede usarse una clave mnemónica (véase 7.1.7).

Después de la validación de una opción de menú, se ejecutará la acción local asociada y se cerrarán el menú en cascada y el menú desplegable (menú superpuesto).

Funcionalidad interactiva

- cambio;
- validación.

Aspecto visual (véanse las Figuras 13 y 14)

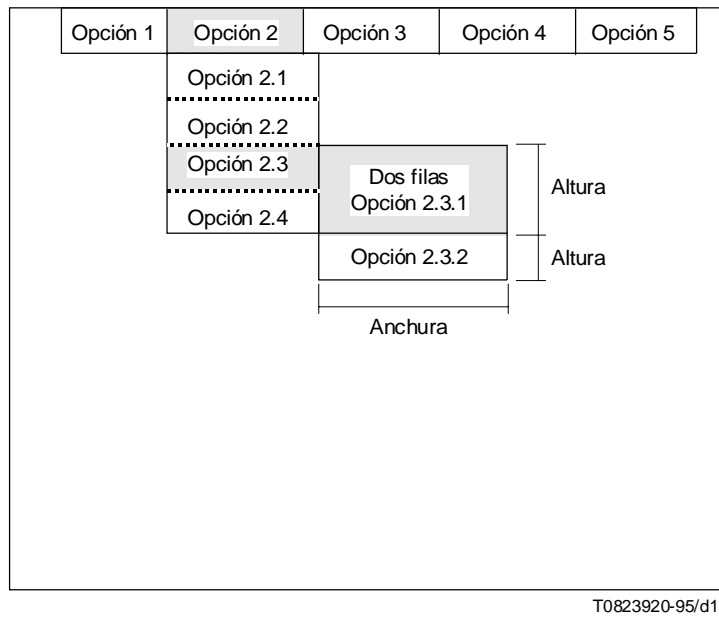


FIGURA 13/T.107
Menú en cascada (presentación recomendada)

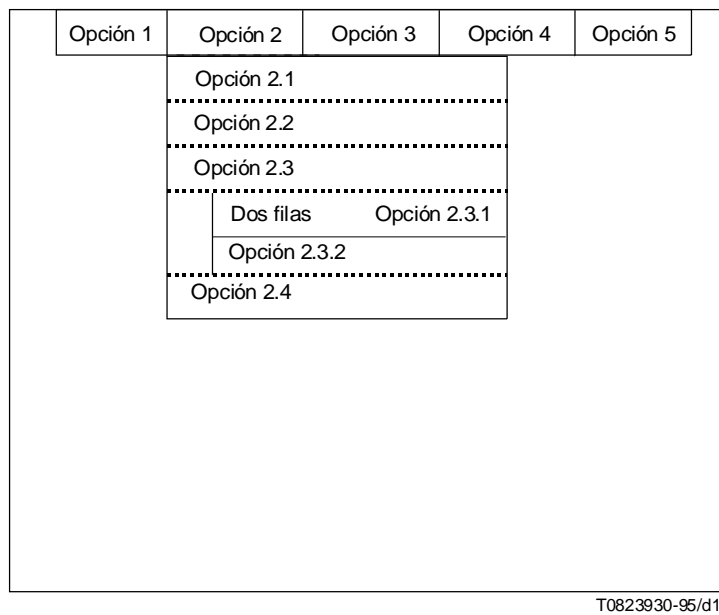


FIGURA 14/T.107
Menú en cascada (presentación sustitutiva)

Atributos

- CIN: Este atributo lleva el número de identificación de componente (véase 7.3, Nota 2).
- Height: Este atributo lleva la altura del componente (véase 7.3, Nota 2).
- Width: Este atributo lleva la anchura del componente. Será igual a todas las selecciones por menú del menú en cascada (véase 7.3, Nota 2).
- NotAccessible: El elemento no estará accesible.

- Text: Este atributo lleva el contenido de texto del componente.
- Separated: Se trazará un separador entre los elementos.
- LocActAct: Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- LocActVal: Este atributo lleva el código de la acción local asociada con el componente y generada por su validación.

7.4 Barra de botones

Descripción

La barra de botones es una lista consecutiva de botones pulsados colocados juntos horizontalmente o verticalmente.

Representación horizontal:

- se asignará el mismo número de filas a cada botón pulsado.

Representación vertical:

- se asignará el mismo número de columnas a cada botón pulsado.

NOTA – Si los atributos «Height» y «Width» están presentes en la definición de una barra de botones, todos los botones tendrán la misma dimensión. Entonces, no estarán presentes los atributos de dimensionamiento de los componentes de la barra de botones. Si en la definición de la barra de botones horizontal sólo está presente el atributo «Height», éste indica la altura de todo el objeto. Entonces, la anchura de los componentes se da con cada componente. Si en la definición de la barra de botones vertical sólo está presente el atributo «Width», éste indica la anchura de todo el objeto. La altura de los componentes se da con cada componente.

Comportamiento

Cuando la barra de botones está activa, el terminal realizará un botón pulsado y ofrecerá al usuario facilidades de cambio y validación.

Funcionalidad interactiva

- cambio;
- validación.

Aspecto visual (véase la Figura 15)

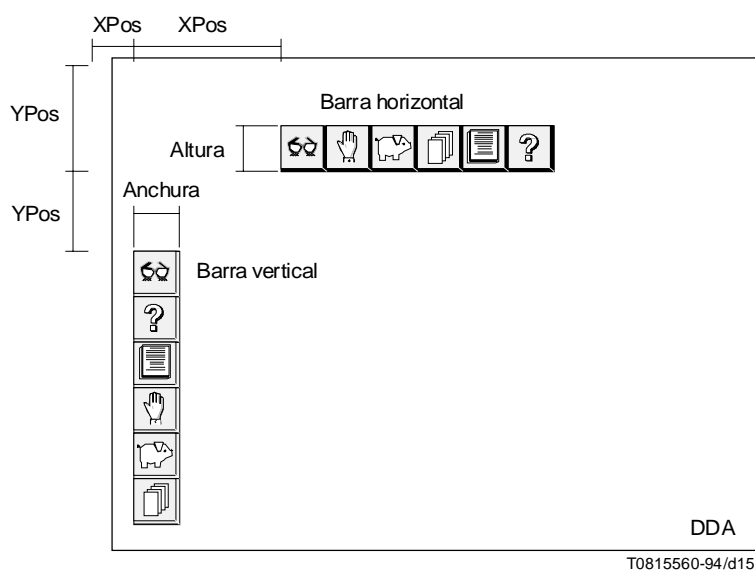


FIGURA 15/T.107

Barra de botones

Atributos

- XPos: Este atributo lleva la posición horizontal del elemento en la NDC.
- YPos: Este atributo lleva la posición vertical del elemento en NDC.
- Vertical: Conmutador booleano entre las alternativas.
- Height: Este atributo lleva la altura de la barra de botones. Sólo es aplicable a una barra de botones horizontal (véase la Nota).
- Width: Este atributo lleva la anchura de la barra de botones. Sólo es aplicable a una barra de botones vertical (véase la Nota).
- FirstActive: Este atributo lleva el CIN de la barra de botones, que está activa por defecto la primera vez que se abre el objeto.
- Modal: El elemento será modal.
- Closed: El elemento estará en estado cerrado.
- NotAccessible: El elemento no estará accesible.

7.4.1 Composición

7.4.1.1 Componente botón

Descripción

El botón es una zona rectangular en la DDA.

Si está formado por datos de texto, debe representarse mediante una etiqueta de texto, posiblemente asociada con un gráfico para dibujar la forma del botón. La forma gráfica del botón se incluirá en el espacio asignado por la aplicación VEMMI para todo el componente botón. Si la forma gráfica no cubre todo el espacio asignado para el botón, el terminal puede redimensionar el contenido. El terminal puede centrar los datos de texto en la zona de visualización de elementos reservada por la aplicación VEMMI para el componente.

El dibujo del botón y el posible efecto generador visual dependen del terminal.

Comportamiento

Cuando un botón pulsado está activo, el terminal lo realzará. Para validar el botón puede usarse una clave mnemónica (véase 7.1.7).

Cuando se implementa un efecto de activación, el botón volverá al estado de visualización inicial después de la validación de usuario.

Funcionalidad interactiva

- activación;
- validación.

Aspecto visual (véase la Figura 16)

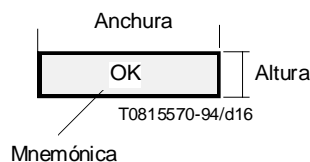


FIGURA 16/T.107

Botón

Atributos

- CIN: Este atributo lleva el número de identificador de componente (CIN).
- Height: Este atributo lleva la altura del componente. Sólo es aplicable a botones pulsados de barras de botones verticales (véase 7.4, Nota).
- Width: Este atributo lleva la anchura del componente. Sólo es aplicable a botones pulsados de barras de botones horizontales (véase 7.4, Nota).
- Closed: El elemento estará en el estado cerrado.
- NotAccessible: El elemento no estará accesible.
- BIN: El BIN del componente.
- Text: El contenido de texto del componente.
- LocActAct: Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- LocActVal: Este atributo lleva el código de la acción local asociada con el componente y generada por su validación.

7.5 Menú superpuesto

El menú superpuesto se subdivide en dos diferentes grupos lógicos de componentes de opción de menú. Estos grupos difieren en su comportamiento y funcionalidad. Los dos grupos diferentes se denominan:

- menú superpuesto primario;
- menú en cascada.

El menú superpuesto primario es una lista vertical de componentes de opción de menú. El menú en cascada es una lista vertical de componentes de selección por menú asociados con el mismo componente de opción de menú del menú superpuesto primario.

Puede utilizarse un separador de opciones de menú para separar visualmente componentes de opciones de menú en un grupo lógico. Se define fijando el atributo «separador» con la definición de una opción de menú ordinaria.

La estructura del objeto menú superpuesto viene dada, con los siguientes convenios, por el orden descendente de sus componentes (trayecto de arriba a abajo).

En el ejemplo de menú superpuesto siguiente, el orden descendente canónico de los componentes es:

(BFG(IJK)H)

B	
F	
G	I
H	J
	K

NOTA 1 – Los paréntesis del ejemplo anterior sólo sirven para mostrar los diferentes niveles de encapsulado de los diferentes grupos de objetos para el trayecto de descripción.

La descripción de la estructura del objeto es entonces:

- Objeto: Menú superpuesto.
- Componente: Opción de menú superpuesto B.
- Componente: Opción de menú superpuesto F.
- Componente: Opción de menú superpuesto G.
- Componente: Opción de menú en cascada I.

- Componente: Opción de menú en cascada J.
- Componente: Opción de menú en cascada K.
- Componente: Opción de menú superpuesto H.

Funcionalidades interactivas

- cambio;
- validación.

Aspecto visual general (véase la Figura 17)

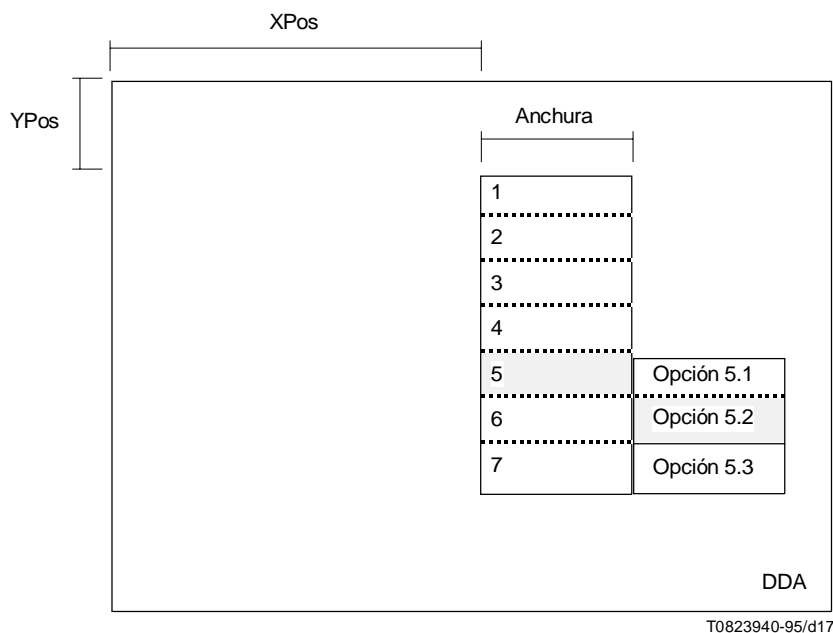


FIGURA 17/T.107

Puede darse un título al menú superpuesto, que será visualizado en su primera fila del menú superpuesto primario.

NOTA 2 – Todos los atributos de dimensionamiento de los componentes de opción de menú del menú superpuesto son opcionales. Si los atributos de dimensionamiento se definen en la codificación de un menú superpuesto, el terminal podrá ignorarlos y dimensionar el menú conforme a los requisitos de espacio del contenido de texto. No se enviará ningún mensaje de error al anfitrión.

Atributos

- **XPos:** Este atributo lleva la posición horizontal del elemento en la NDC.
- **YPos:** Este atributo lleva la posición vertical del elemento en NDC.
- **Width:** Este atributo lleva la anchura del menú superpuesto (véase la Nota 2).
- **Title:** Este atributo lleva el título del objeto. El título será visualizado en la primera fila del objeto.
- **TitleFont:** Este atributo lleva el FIN del título.
- **FirstActive:** Este atributo lleva el CIN del componente, que está activo por defecto la primera vez que se abre el objeto.
- **Modal:** El elemento será modal.
- **Closed:** El elemento estará en el estado cerrado.
- **NotAccessible:** El elemento no estará accesible.

7.5.1 Composición

7.5.1.1 Componentes de opción de menú del menú superpuesto primario

Descripción

El menú superpuesto primario es una lista consecutiva de opciones de menú y se presenta verticalmente en varias filas.

Se asignará el mismo número de columnas a cada opción de menú.

Puede utilizarse un separador en el menú superpuesto primario para separar visualmente diferentes opciones de menú.

Comportamiento

Cuando el menú superpuesto está activo, el terminal realzará una opción de menú, y ofrecerá al usuario facilidades de cambio y validación. Para validar la opción de menú puede usarse una clave mnemónica (véase 7.1.7).

Si se valida un componente menú superpuesto, se ejecutará la acción local asociada y luego se abrirá el menú en cascada correspondiente (el componente menú superpuesto permanece realzado). Si un menú en cascada no está asociado con una opción de menú después de una validación de usuario, se ejecutará la acción local correspondiente y se cerrará el menú superpuesto.

Funcionalidad interactiva

- cambio;
- validación.

Aspecto visual (véase la Figura 18)

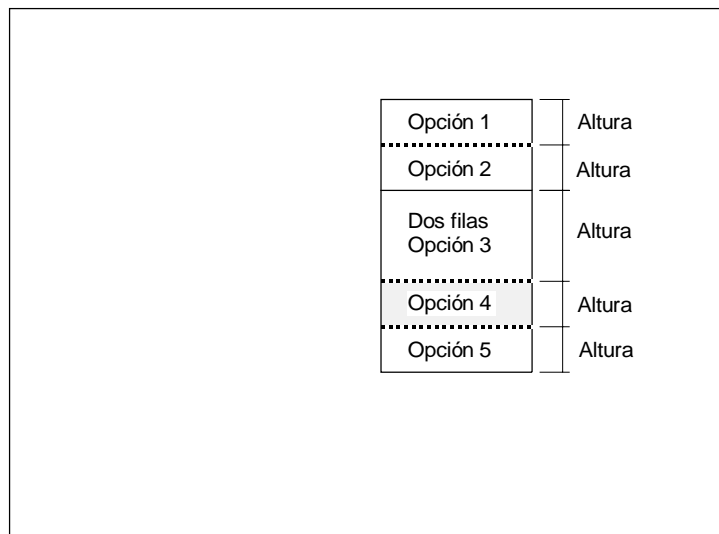


FIGURA 18/T.107

Menú superpuesto

Atributos

- CIN: Este atributo lleva el número de identificación de componente.
- Height: Este atributo lleva la altura del componente (véase 7.5, Nota 2).
- Closed: El elemento estará en estado cerrado.

- NotAccessible: El elemento no estará accesible.
- Text: Este atributo lleva el contenido de texto del componente.
- Separated: Los elementos estarán separados.
- LocActAct: Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- LocActVal: Este atributo lleva el código de la acción local asociada con el componente y generada por su validación.

7.5.1.2 Componentes de opción de menú del menú en cascada

Descripción, Comportamiento, Funcionalidad interactiva, Atributos:

Véase 7.3.1.3.

Aspecto visual (véase la Figura 19)

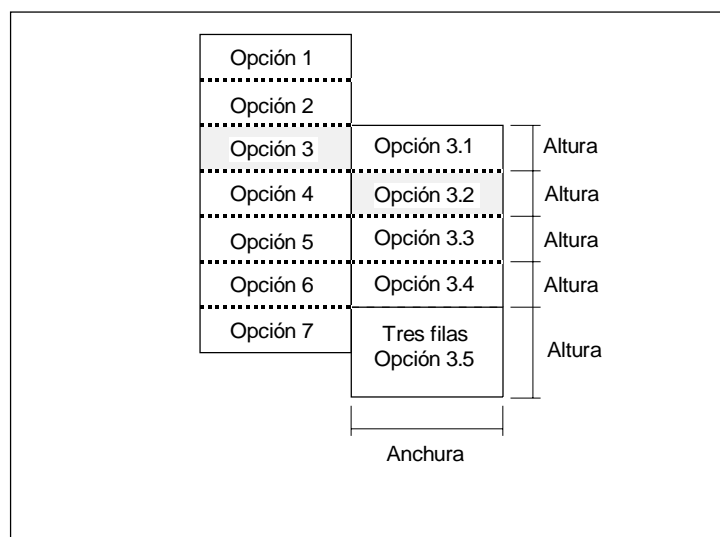


FIGURA 19/T.107

Presentación recomendada de un menú superpuesto con el menú en cascada asociado

7.6 Casilla de diálogo

Descripción

La casilla de diálogo es una zona rectangular en la DDA que contiene componentes VEMMI para establecer la interacción de usuario.

Los componentes contienen (o tienen referencias a) otros componentes u objetos recurso. La Figura 20 muestra esa relación. Cada flecha indica una posible referencia a un componente o a un objeto recurso. Las referencias pueden ser múltiples, y la caja de diálogo puede contener varios componentes del mismo tipo.

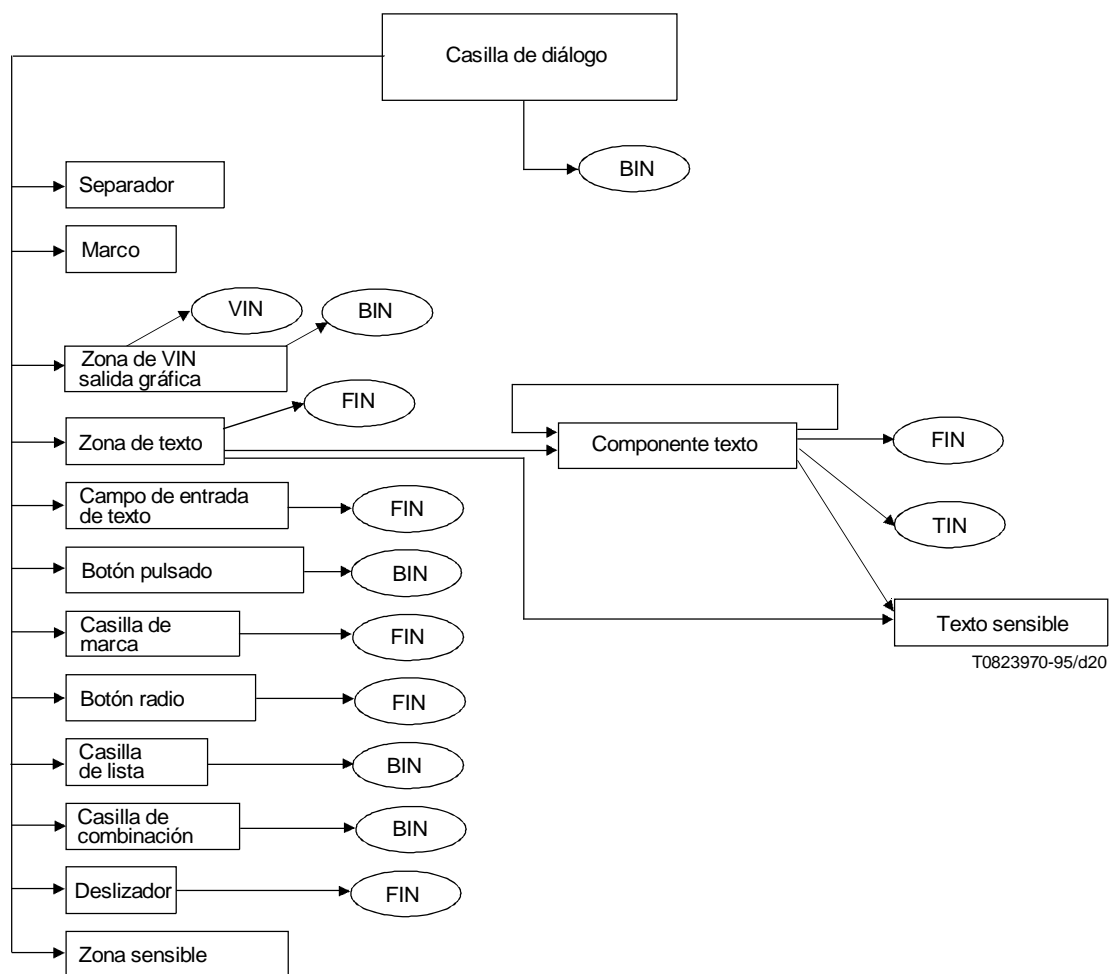


FIGURA 20/T.107

Relaciones de elemento dentro de la casilla de diálogo

Se puede dar un título de texto a la casilla de diálogo que será visualizado en su primera fila. La altura del título es igual a 1/32 NDC.

La casilla de diálogo puede dotarse de una zona límite que debe ser igual a una posición de carácter. Cuando está presente, esta zona se incluirá en las dimensiones definidas por la aplicación VEMMI. Cuando se solicita un límite, el terminal dibujará un marco. Este dibujo del marco depende del terminal (no rebasará un ancho de 1/160 NDC).

El objeto origen de la casilla de diálogo es la esquina superior izquierda del rectángulo de la casilla, sin tener en cuenta si la casilla tiene o no un título. Los componentes se colocan con respecto a este punto cuando las casillas de diálogo no llevan título. En el caso de casillas con título de texto, se desplaza este punto hacia abajo una distancia correspondiente a la altura de los títulos y a la anchura de la zona límite, y horizontalmente una distancia igual a la anchura de la zona lateral.

La zona de fondo de la casilla de diálogo es el rectángulo definido por las dimensiones de la casilla, salvo la barra de título y el límite. El fondo puede estar coloreado uniformemente o tener un contenido de datos.

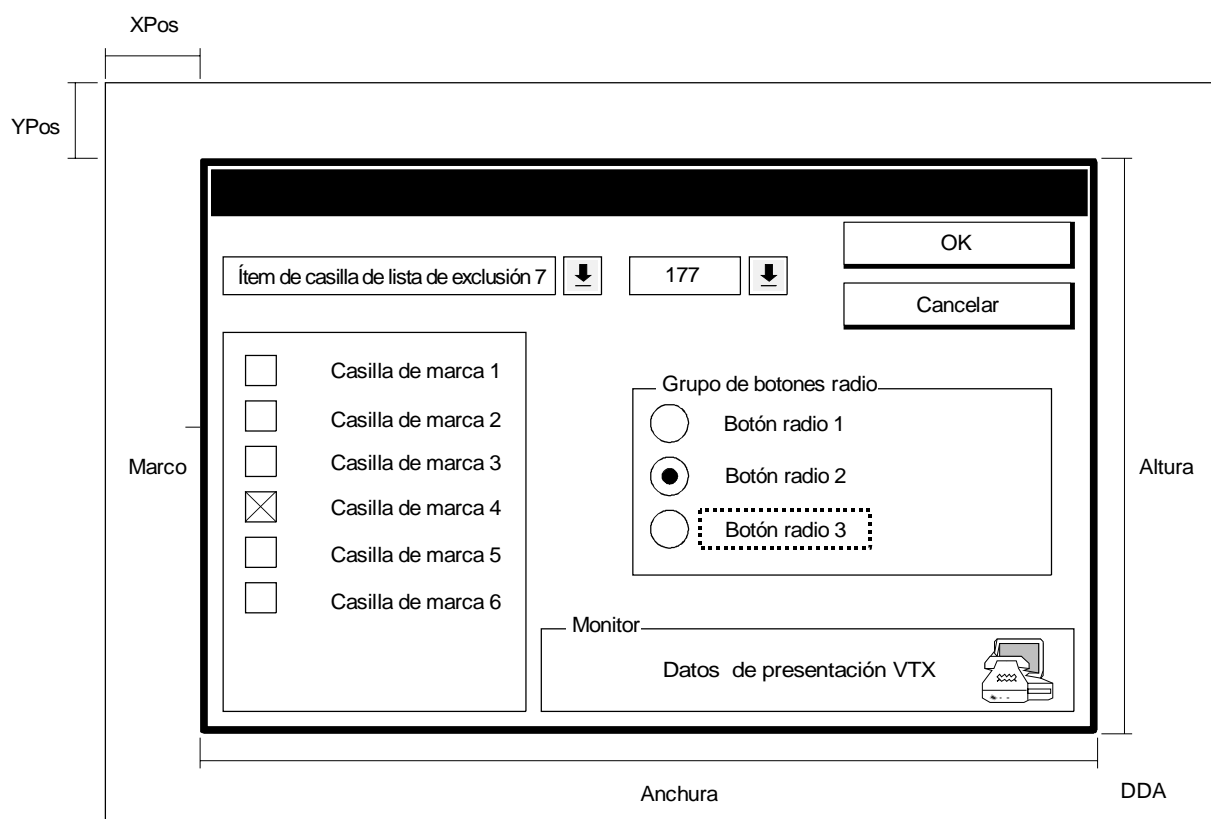
Comportamiento

El usuario puede activar cualquier componente accesible con un dispositivo de apuntamiento o con el teclado. Si el usuario activa los componentes con las teclas que proporcionan la funcionalidad «Previo» y «Siguiente», el orden de activación corresponderá al orden de recepción de los componentes de la casilla de diálogo.

Funcionalidad interactiva

- Desplazamiento.

Aspecto visual (véase la Figura 21)



T0815610-94/d21

FIGURA 21/T.107

Caja de diálogo

Atributos

- XPos: Este atributo lleva la posición horizontal del elemento en NDC.
- YPos: Este atributo lleva la posición vertical del elemento en NDC.
- Width: Este atributo lleva la anchura del objeto.
- Height: Este atributo lleva la altura del objeto.
- NoBorder: No se trazará ningún límite.
- Title: Este atributo lleva el título del objeto. El título se visualizará en la primera fila del objeto.

- **FirstActive:** Este atributo lleva el CIN del componente, que está activo por defecto la primera vez que se abre el objeto.
- **Modal:** El elemento será modal.
- **Closed:** El elemento estará en estado cerrado.
- **NotAccessible:** El elemento no estará accesible.
- **StoreInitialValues:** El terminal almacenará los valores iniciales del componente en el momento de la creación del objeto. Si se aplica la acción local «restablecer valores iniciales», esos valores iniciales se restablecerán.
- **Maximisable:** El objeto es maximizable.
- **Colour:** Índice de color del fondo.
- **BIN:** Número de identificación de mapa de bits para rellenar el fondo de la casilla de diálogo.
- **DispType:** Especifica el modo de visualización del mapa de bits de fondo: centrado, elástico (por defecto) o en Posado (tiled).

7.6.1 Composición

7.6.1.1 Componente separador

Descripción

El separador es una línea de trazo continuo horizontal o vertical utilizada para separar diferentes zonas dentro de una casilla de diálogo.

Comportamiento

El separador será inaccesible.

Funcionalidades interactivas

- Ninguna.

Aspecto visual (véase la Figura 22)

Atributos

- **CIN:** Este atributo lleva el número de identificación de componente.
- **XPos:** Este atributo lleva la posición horizontal del elemento en NDC.
- **YPos:** Este atributo lleva la posición vertical del elemento en NDC.
- **Vertical:** El elemento se representará vertical.
- **Height:** Este atributo lleva la altura del componente. Sólo es aplicable a separadores verticales.
- **Width:** Este atributo lleva la anchura del componente. Sólo es aplicable a separadores horizontales.
- **Closed:** El elemento estará en el estado cerrado.
- **Colour:** Índice de color.

7.6.1.2 Componente marco

Descripción

Un marco consiste en cuatro líneas de trazo continuo que separan visualmente una zona rectangular de la casilla de diálogo.

Comportamiento

El marco será inaccesible.

Funcionalidad interactiva

- Ninguna.

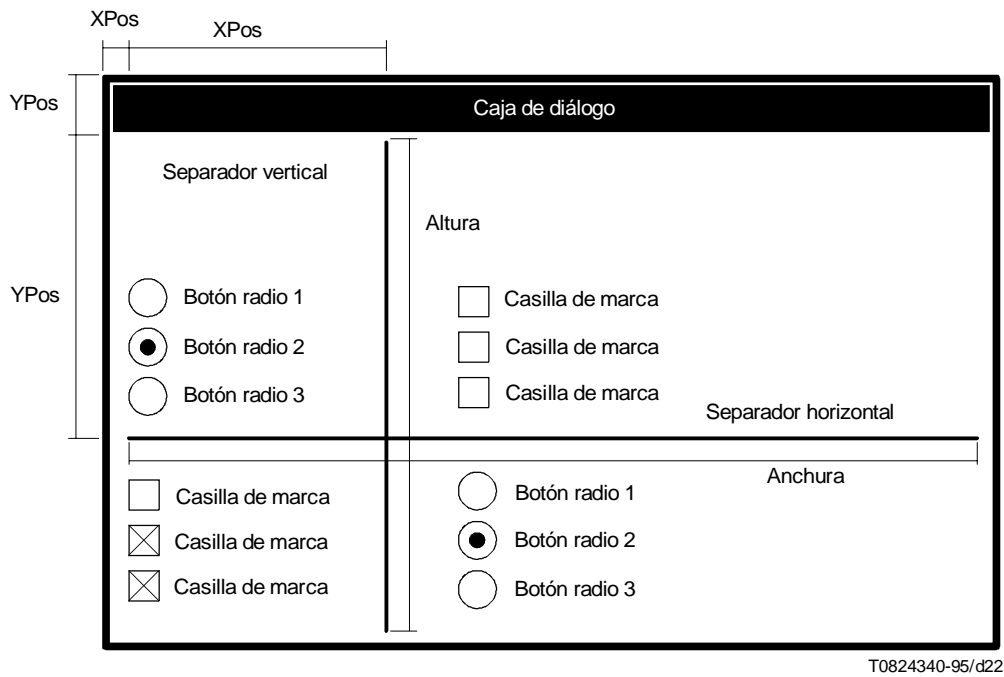


FIGURA 22/T.107
Separador

Aspecto visual (véase la Figura 23)

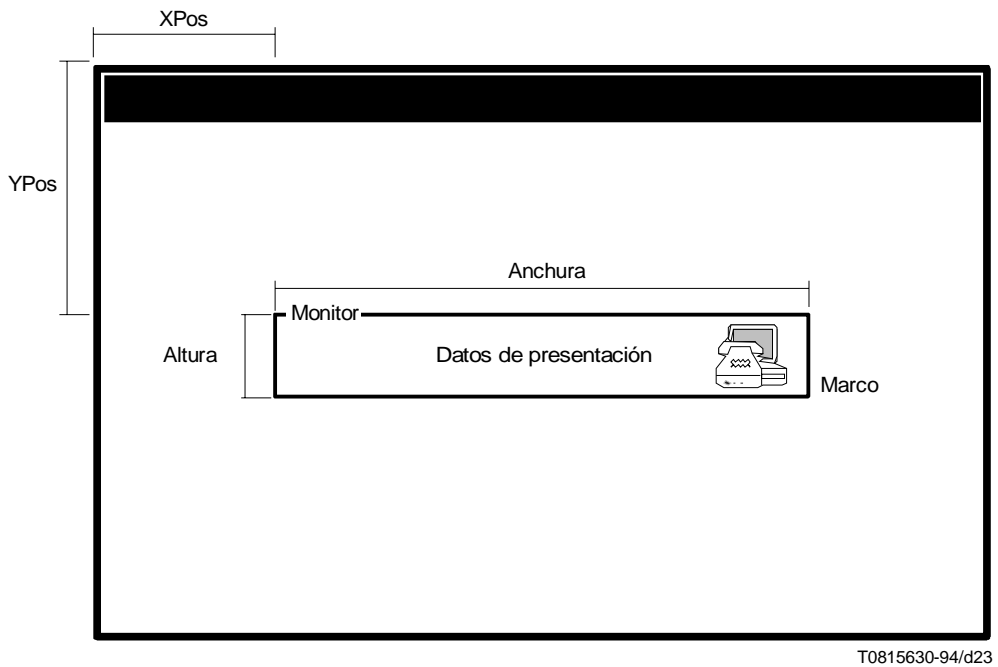


FIGURA 23/T.107
Marco titulado con una zona de presentación de texto

Atributos

- CIN: Este atributo lleva el número de identificación de componente.
- XPos: Este atributo lleva la posición horizontal del elemento en NDC.
- YPos: Este atributo lleva la posición vertical del elemento en NDC.
- Width: Este atributo lleva la anchura del componente.
- Height: Este atributo lleva la altura del componente.
- Closed: El elemento estará en el estado cerrado.
- Colour: Índice de color.

7.6.1.3 Componente zona de presentación de texto

Descripción

La zona de presentación de texto es una zona destinada a presentar datos de texto al usuario. Básicamente, este elemento se usa para presentar texto que cabe en una ventana (el espacio definido por un componente zona de texto), o para presentar datos de texto grande, en los que el usuario tiene que emplear herramientas de desplazamiento para navegar en el texto.

La zona de presentación de texto tiene su punto de comienzo en la esquina superior izquierda del rectángulo definido por el componente, y el texto se visualiza línea por línea, de izquierda a derecha. El contenido de texto puede contener atributos «in-text» para cambiar los diferentes tipos de caracteres y para definir las «cadenas de texto sensible». Puede haber varios atributos «in-text», que aparecerán en cualquier orden; se los evalúa secuencialmente durante la operación de visualización y sobre la base de la información estructural que contienen los componentes de texto (atributos «previous, next»).

Desplazamiento, Límite, Título

Cuando el espacio necesario para la presentación de los datos es mayor que el asignado al componente, la aplicación de terminal proporcionará herramientas de desplazamiento vertical. El terminal puede presentar:

- herramientas de desplazamiento próximas al lado derecho de la zona;
- herramientas de desplazamiento reducidas como botones superpuestos en la zona;
- teclas de cursor que suministran la funcionalidad de desplazamiento.

El espacio necesario para las herramientas de desplazamiento está incluido en las dimensiones generales del componente.

El componente puede tener un límite. Si está presente, el límite estará incluido en las dimensiones generales del componente. Cuando se solicita un límite, el terminal dibujará un marco. Ese dibujo de un marco depende del terminal. No será superior a 1/160 NDC. Si se proporcionan herramientas de desplazamiento próximas al lado derecho del componente, el límite se ampliará para incluirlas visualmente en la casilla.

Operación maximizar

El usuario puede cambiar el tamaño de la casilla de diálogo con las herramientas de desplazamiento proporcionadas por la interfaz de usuario gráfica (GUI) y mediante un botón de maximizar definido. La estrategia recomendada se describe en 7.1.5; se trata de recomendaciones por defecto. Si para el componente se elige el atributo «NoFormat», se aplicará lo siguiente:

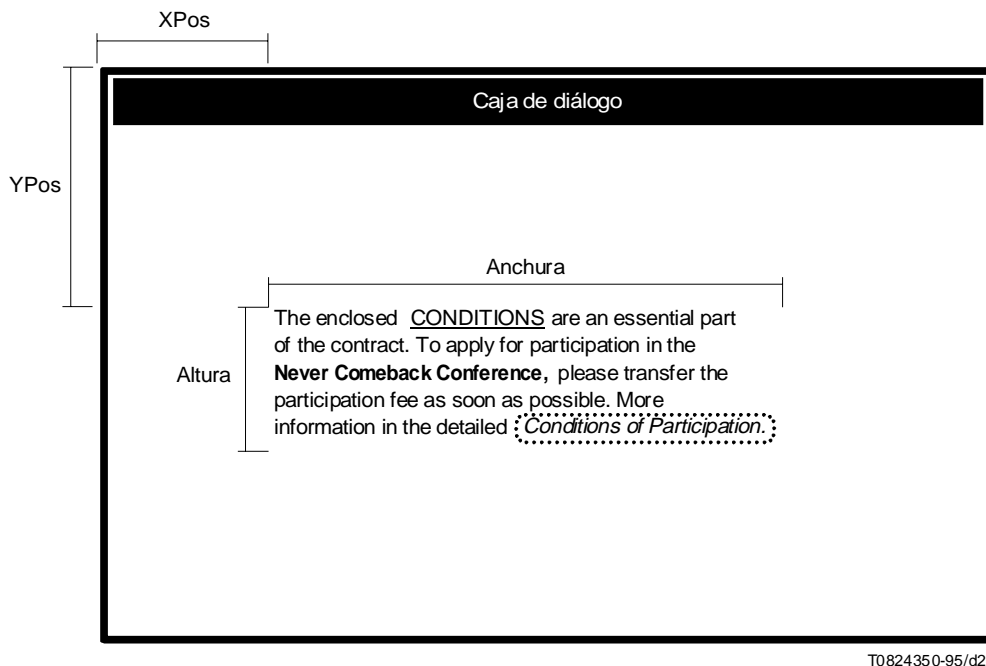
- un aumento vertical no debe cambiar la forma del texto; no debe añadirse texto adicional;
- una disminución vertical debe cortar el texto a la altura del nuevo límite;
- un aumento horizontal debe añadir nuevas líneas de texto;
- una disminución horizontal debe cortar el texto.

Funcionalidad interactiva

- Activación y validación del texto sensible, definido mediante atributos in-text.

NOTA – Para transferir un gran volumen de datos, se recomienda no utilizar la definición de texto directo, sino utilizar el objeto recurso de texto y transmitir los datos utilizando la transferencia de recursos VEMMI.

Aspecto visual (véase la Figura 24)



T0824350-95/d24

FIGURA 24/T.107
Zone de presentación de texto

Atributos

- CIN: Este atributo lleva el número de identificación de componente.
- XPos: Este atributo lleva la posición horizontal del elemento en NDC.
- YPos: Este atributo lleva la posición vertical del elemento en NDC.
- Width: Este atributo lleva la anchura del componente.
- Height: Este atributo lleva la altura del componente.
- NoScrollingTools: El terminal no representará herramientas de desplazamiento.
- Format: Una maximización de la caja de diálogo usará las reglas de formatación limitada antes descritas.
- NoBorder: No se trazará ningún límite para enmarcar el elemento.
- InitialFnt: FIN del tipo de caracteres de atributo que se aplicará al primer texto.

- **AutoScroll:** Modificación de un componente referenciado por la zona de texto que provoca una visualización inmediata de ese componente modificado.
- **InText:** Contenido de texto que posiblemente incluye atributos dentro del texto y referencias a objetos tipo de caracteres.
- **TextCompRef:** Referencia a un componente texto.
- **Closed:** El elemento estará en el estado cerrado.

7.6.1.4 Componente texto

Descripción

Este componente se utiliza para separar datos de texto grande en unidades (componentes de texto), y para definir la información estructural necesaria (concatenación de los componentes de texto) para su visualización en una zona de texto. Esa información estructural consiste en referencias a los componentes de texto previos y siguientes, definidos por los atributos «NextText» y «PreviousText».

Concepto de visualización

Los componentes no tienen necesariamente la misma longitud. Durante la visualización de componentes de texto concatenados, el usuario tendrá la impresión de ver un texto continuo. La continuidad no se interrumpirá por un cambio entre componentes. Con los componentes consecutivos se ofrecerán capacidades de desplazamiento vertical, línea por línea y página por página. La continuidad de la visualización se interrumpe en dos casos:

- el componente no contiene otra concatenación, no hay «NextText» durante el desplazamiento hacia adelante, o «PreviousText» durante el desplazamiento hacia atrás. La visualización no continúa porque el texto recientemente visualizado pertenece a un tema que ha alcanzado su fin lógico. La Figura 29 muestra dos secuencias de componentes concatenados: los CIN 7, 8 y 9, y CIN 17;
- «saltos» ocasionados por la validación de un texto sensible como parte de los componentes visualizados u otros componentes interactivos de la caja de diálogo. El atributo de texto sensible puede contener, como una acción local, una instrucción específica de elemento, por ejemplo, «abrir el componente del objeto progenitor». El componente progenitor referenciado es un componente de texto, que será visualizado en la posición de comienzo y no consecutivamente con respecto al último componente visualizado. Este efecto se ilustra en las Figuras 27 y 28.

El texto a visualizar está contenido en la definición de componente, en otro componente de texto referenciado o en un objeto recurso referenciado. Si el terminal no dispone de un objeto recurso de ese tipo, puede solicitarse a la aplicación anfitriona con la instrucción VEMMI_Retransmisión_objeto y se aplicará la regla B (véase 5.3.2). Esa petición puede ser iniciada debido a un desplazamiento hacia adelante para buscar datos que aún no están en el terminal, pero en general no existe una relación directa entre la acción de desplazamiento y la petición de retransmisión de objeto porque el terminal mantiene los objetos ya recibidos.

NOTA – Para transferir un gran volumen de datos se recomienda no utilizar la definición de texto directo, sino utilizar el objeto recurso de texto y transmitir los datos utilizando la transferencia de recursos VEMMI.

Aspecto visual (véanse las Figuras 25 a 28)

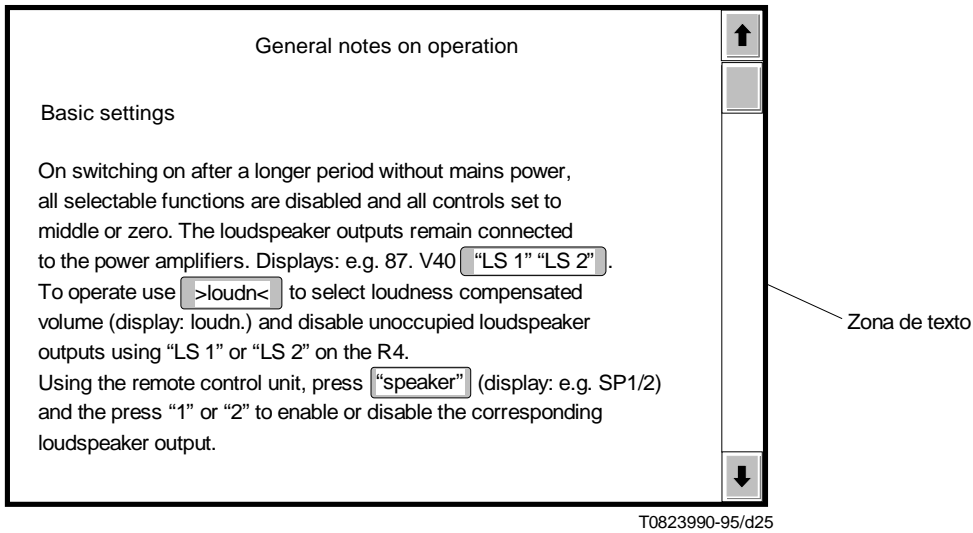


FIGURA 25/T.107

**Una zona de texto con componente(s) de texto, barras de desplazamiento y textos sensibles (el atributo "NoFormat" no está fijado).
A continuación, el usuario actúa y desplaza el botón hacia abajo ...**

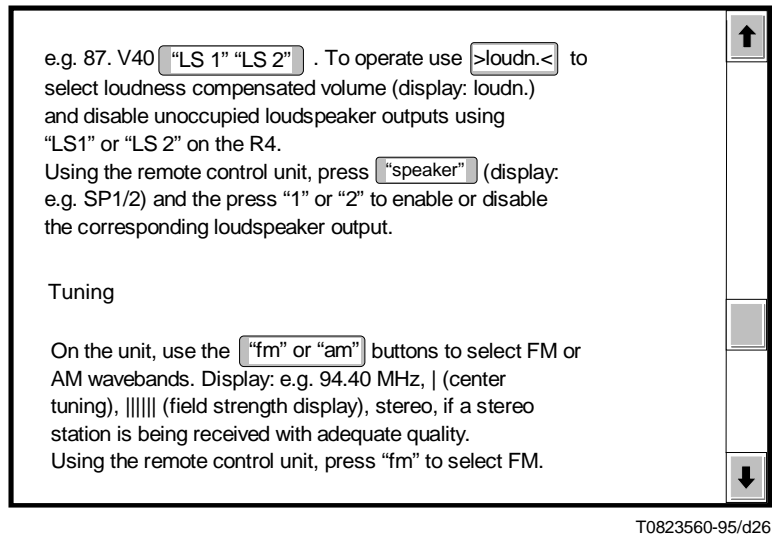
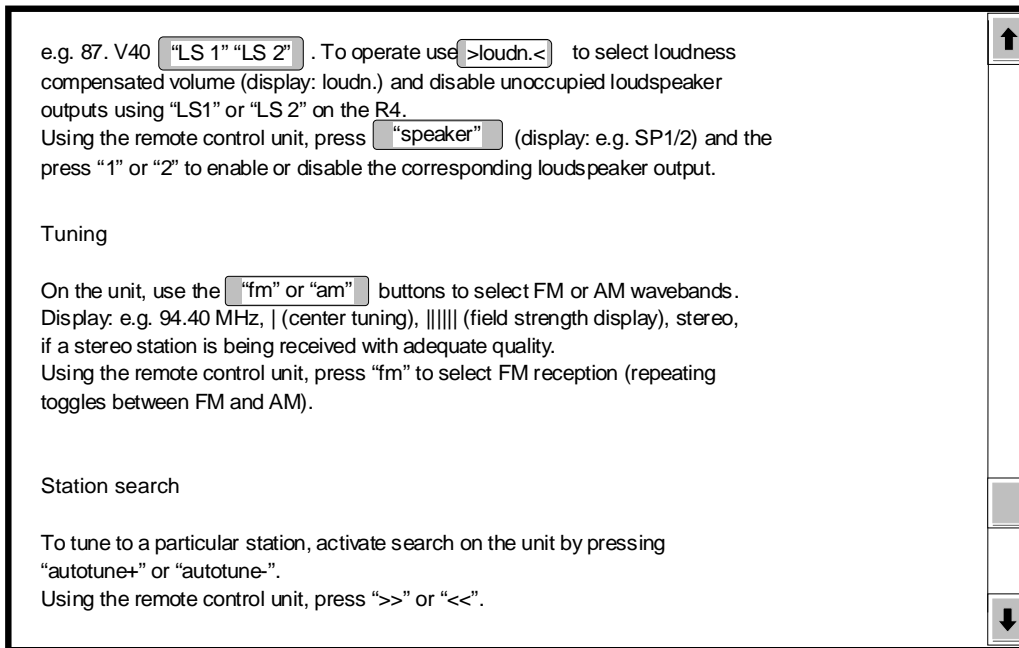


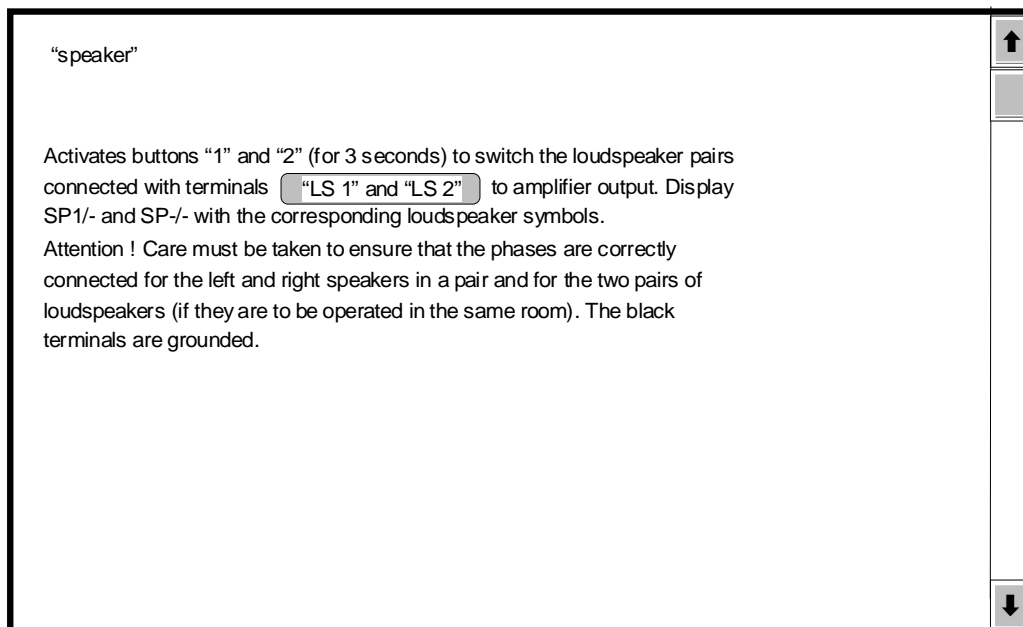
FIGURA 26/T.106

... además, agranda la ventana en ambos sentidos, ...



T0823570-95/d27

FIGURA 27/T.107
 ... y por último, hace un click en el hot spot "speaker"



T0823580-95/d28

FIGURA 28/T.107
 El texto "speaker" es visualizado

Una posible arquitectura de componente para la secuencia de visualización de las Figuras 25 a 29 puede ser la siguiente (no se enumeran los atributos in-text):

<p>General notes on operation</p> <p>Basic settings</p> <p>On switching on after a longer period without mains power, all selectable functions are disabled and all controls set to middle or zero. The loudspeaker outputs remain connected to the power amplifiers. Displays: e.g. 87. V40 "LS 1" "LS 2". To operate use >loudn.< to select loudness compensated volume (display: loudn.) and disable unoccupied loudspeaker outputs using "LS 1" or "LS 2" on the R4.</p> <p>Using the remote control unit, press "speaker" (display: e.g. SP1/2) and the press "1" or "2" to enable or disable the corresponding loudspeaker output.</p>	<p>Componente texto</p> <p>CIN: 7</p> <p>Texto siguiente: 8</p>
<p>Tuning</p> <p>On the unit, use the "fm" or "am" buttons to select FM or AM wavebands. Display: e.g. 94.40 MHz, (centre tuning), (field strength display), stereo, if a stereo station is being received with adequate quality.</p> <p>Using the remote control unit, press "fm" to select FM reception (repeating toggles between FM and AM).</p>	<p>Componente texto</p> <p>CIN: 8</p> <p>Texto previo: 7</p> <p>Texto siguiente: 9</p>
<p>Station search</p> <p>To tune to a particular station, activate search on the unit by pressing "autotune+" or "autotune-".</p> <p>Using the remote control unit, press ">>" or "<<".</p> <p>The search, with automatic muting, continues in the selected direction until the next station with adequate reception quality and centre tuning position is located.</p>	<p>Componente texto</p> <p>CIN: 9</p> <p>Texto previo: 8</p>
<p>"speaker"</p> <p>Activates buttons "1" and "2" (for 3 seconds) to switch the loudspeaker pairs connected with terminals "LS 1" and "LS 2" to amplifier output. Display SP1/- and SP-/- with the corresponding loudspeaker symbols.</p> <p>Attention! Care must be taken to ensure that the phases are correctly connected for the left and right speakers in a pair and for the two pairs of loudspeakers (if they are to be operated in the same room). The black terminals are grounded.</p>	<p>Componente texto</p> <p>CIN: 17</p>

FIGURA 29/T.107

Arquitectura del componente texto

La interactividad local puede implementarse mediante la cooperación adecuada de las acciones locales (parámetro LocActVal) que forman parte del texto sensible y de otros elementos activos de la casilla de diálogo. En el caso del ejemplo, el usuario puede ir del texto de la Figura 27 al de la Figura 25 mediante el botón «Comenzar», si el botón contiene la acción local «abrir componente 7».

Atributos

- CIN: Número de identificación de componente.
- PreviousText: CIN del componente de texto concatenado hacia atrás.
- CurrentText: Definición de texto, incluidas las referencias a los objetos recurso tipo de letra, objeto recurso texto y componentes de texto sensible.
- NextText: CIN del componente de texto concatenado hacia adelante.

7.6.1.5 Componente texto sensible

Este componente define las operaciones de activación y validación de las cadenas de texto sensible, que forman parte de los componentes de texto.

Aspecto visual

Véase 7.6.1.4.

Atributos

- CIN: Número de identificación de componente.
- NotAccessible: El elemento no estará accesible.
- LocActAct: Especifica las acciones locales asociadas con el componente y generadas por su activación.
- LocActVal: Especifica las acciones locales asociadas con el componente y generadas por su validación.

7.6.1.6 Componente zona de salida gráfica

Descripción

La zona de salida gráfica es un rectángulo para visualizar datos de salida gráfica (mapas de bits), datos de videotex, etc. El mapa de bits está referenciado por un BIN. Hay tres modos de visualización diferentes: el mapa de bits puede estar centrado en la zona de salida gráfica, puede estar agrandado de forma que cubra toda la zona o puede estar enlosado (tiled). En cualquiera de los casos, el mapa de bits no rebasará el espacio reservado con este componente. Si el mapa de bits referenciado no existe mientras el componente está abierto, no tiene ningún efecto visual.

Los datos videotex son referenciados mediante un VIN. En cualquier caso, los datos videotex no rebasarán el espacio reservado con este componente. Si un contenido de datos referenciado no existe mientras el componente está abierto, no tiene efecto visual.

Comportamiento

Es inaccesible.

Funcionalidades interactivas

- Ninguna.

Aspecto visual (véase la Figura 30)

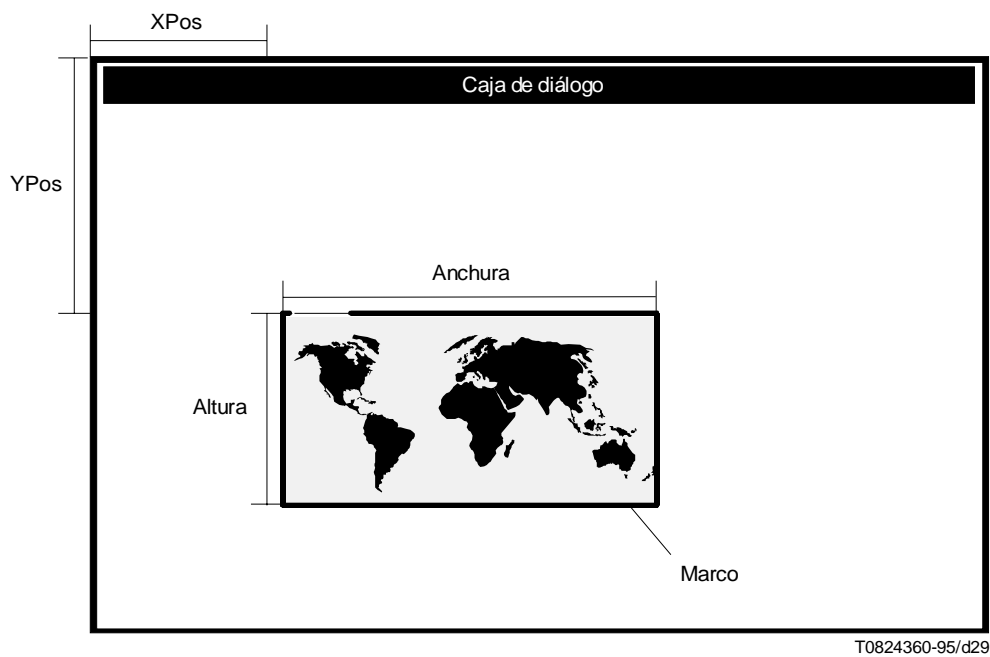
Atributos

- CIN: Número de identificación de componente
- XPos: Posición horizontal de la esquina superior izquierda del rectángulo de mapa de bits.
- YPos: Posición vertical de la esquina superior izquierda del rectángulo de mapa de bits.
- Width: Anchura del rectángulo de mapa de bits.
- Height: Altura del rectángulo de mapa de bits.
- Closed: El elemento estará en el estado cerrado.
- DispType: Especifica el modo de visualización del mapa de bits; centrado, extendido (por defecto) o enlosado (tiled).
- BIN: Número de identificación de mapa de bits.
- VIN: Número de identificación videotex.

7.6.1.7 Componente botón pulsado

Descripción, Comportamiento, Funcionalidad Interactiva

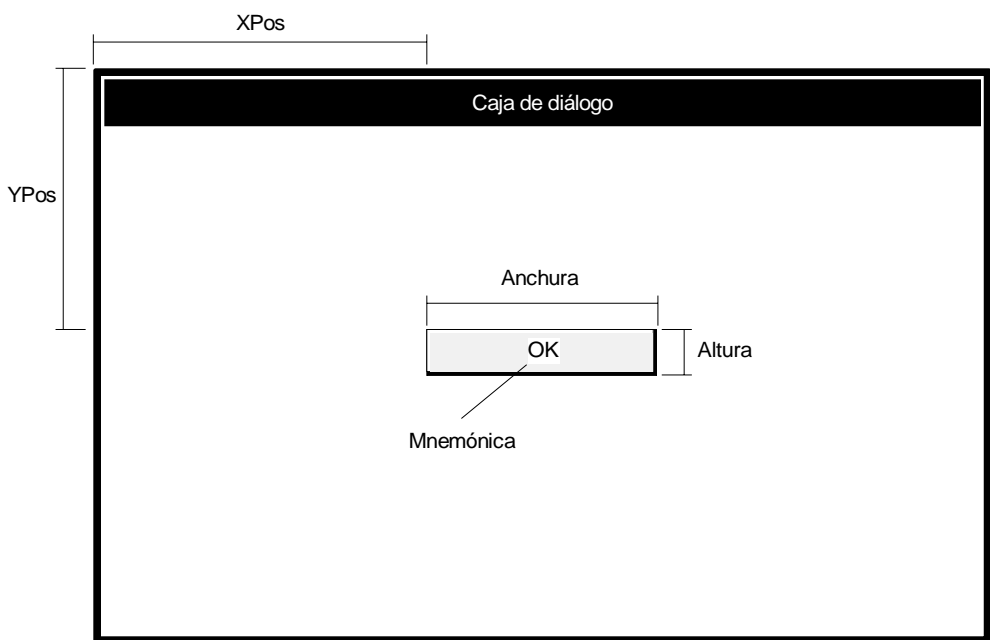
Véase 7.4.1.1.



T0824360-95/d29

FIGURA 30/T.107
Marco y mapa de bits

Aspecto visual (véase la Figura 31)



T0824370-95/d30

FIGURA 31/T.107
Botón pulsado

Atributos

- CIN: Este atributo lleva el número de identificación de componente.
- XPos: Este atributo lleva la posición horizontal del elemento en NDC.
- YPos: Este atributo lleva la posición vertical del elemento en NDC.
- Width: Este atributo lleva la anchura del componente.
- Height: Este atributo lleva la altura del componente.
- Closed: El elemento estará en el estado cerrado.
- NotAccessible: El elemento no estará accesible.
- BIN: El BIN del componente.
- Text: El contenido de texto del componente.
- LocActAct: Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- LocActVal: Este atributo lleva el código de la acción local asociada con el componente y generada por su validación.

7.6.1.8 Componente campo de entrada de texto

Descripción

El campo de entrada de texto se compone de dos ítems:

- la etiqueta de texto;
- la zona de entrada.

Puede asociarse un tipo de caracteres de atributo a la etiqueta de texto. La zona de entrada tiene su punto de comienzo inmediatamente después de la etiqueta de texto. El tipo de los datos de entrada puede ser especificado por la aplicación. Se dispone de los siguientes tipos predefinidos:

- cualquier carácter de texto;
- alfabéticos (A..Z, a..z, caracteres diacríticos);
- numéricos (0..9, +, -, coma, punto, espacio);
- alfanuméricos (alfabéticos, numéricos).

Un atributo del campo de entrada de texto especifica si las entradas de usuario se devuelven o no en eco. La aplicación VEMMI puede definir un carácter para devolver en eco todas las entradas de usuario.

Puede usarse una clave mnemónica para activar el campo de entrada de texto (véase 7.1.7).

Comportamiento

Si el espacio asignado para la zona de entrada en un campo de entrada de texto de una línea no es suficiente para visualizar todos los caracteres introducidos por el usuario, la zona de entrada ofrecerá facilidades de desplazamiento horizontal. Si el espacio asignado para la zona de entrada en un campo de entrada de texto con más de una línea no es suficiente para visualizar todos los caracteres introducidos por el usuario, la zona de entrada ofrecerá facilidades de desplazamiento vertical. Para visualizar todos los posibles caracteres, se recomienda dar espacio suficiente para las entradas de usuario. Si se alcanza el número máximo de los caracteres de entrada, debe indicarse al usuario.

Constricciones

La longitud máxima del texto de la etiqueta está restringida a la primera línea de la anchura del componente campo de entrada de texto. Para los campos de entrada de una línea, la etiqueta se coloca frente a la zona de entrada y se alinea a la izquierda, mientras que la zona de entrada se alinea a la derecha. En el caso de campos de entrada de texto, se coloca en la fila superior y se alinea a la izquierda. No hay espacio reservado para la etiqueta de texto si el atributo TextLabel no está presente.

Funcionalidad interactiva

- activación;
- desplazamiento;
- funcionalidad de edición local de la zona de entrada (dependiente del terminal).

Aspecto visual (véase la Figura 32)

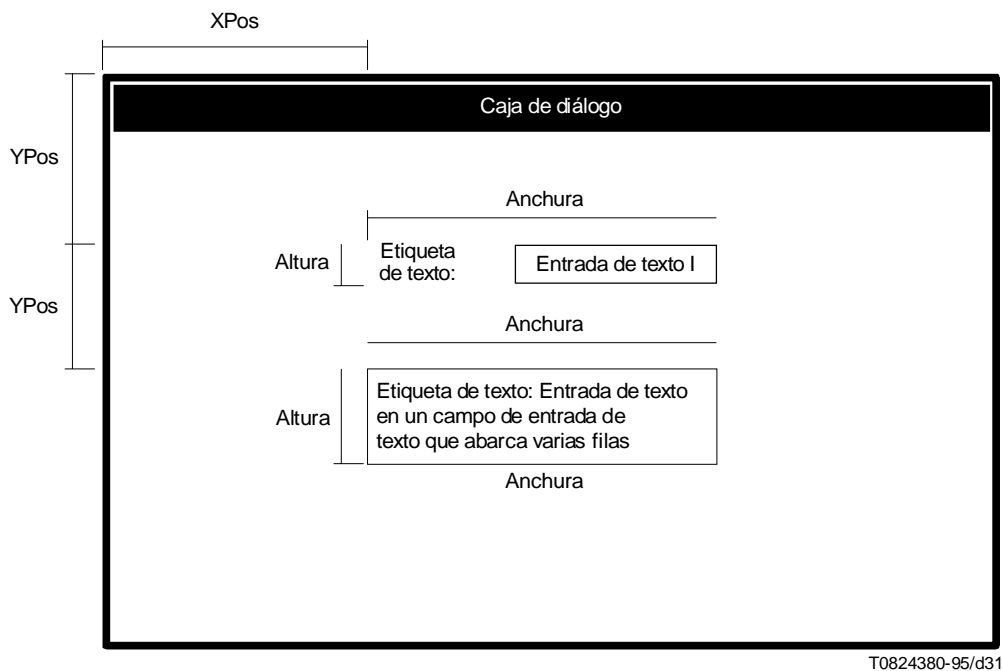


FIGURA 32/T.107

Campo de entrada de texto

Atributos

- CIN: Este atributo lleva el número de identificación de componente.
- XPos: Este atributo lleva la posición horizontal del elemento en NDC.
- YPos: Este atributo lleva la posición vertical del elemento en NDC.
- Width: Este atributo lleva la anchura del componente.
- Height: Este atributo lleva la altura del componente.
- Closed: El elemento estará en el estado cerrado.
- NotAccessible: El elemento no estará accesible.
- MaximTxt: La maximización local puede maximizar el texto del componente proporcionalmente.
- DefValue: Este atributo lleva el valor por defecto del componente.
- TextLabel: Este atributo lleva la etiqueta de texto del componente.
- LabelFont: Especifica el FIN de la etiqueta de texto.
- InputType: Este atributo especifica qué tipos de entrada de usuario aceptará el componente.
- Echo: Este atributo especifica el tipo de eco que se efectuará con las entradas de usuario.
- EchoChar: Este atributo lleva el carácter que se visualizará para devolver en eco las entradas de usuario.

- **MaxChar:** Este atributo lleva el número máximo de caracteres que el usuario podrá introducir en una línea.
- **MaxLine:** Este atributo lleva el número máximo de líneas que el usuario podrá introducir en un campo de entrada multilínea.
- **Multiline:** Indica un campo de entrada multilínea.
- **InputTransformation:** El atributo especifica si el terminal convertirá los caracteres introducidos por el usuario a mayúsculas o a minúsculas, o si no se efectuará ninguna conversión.
- **LocActAct:** Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- **LocActVal:** Este atributo lleva el código de la acción local asociada con el componente y generada por su validación.

7.6.1.9 Componente casilla de marca

Descripción

El componente casilla de marca tiene dos ítems:

- ítem de etiqueta de texto;
- ítem de marca.

Para activar el componente puede usarse una clave mnemónica (véase 7.1.7).

El ítem de marca admitirá dos opciones diferentes, identificables visualmente, que representan los dos posibles valores: marcado o no marcado. El espacio asignado al componente incluirá, en sus dimensiones, el espacio necesario para representar este ítem de marca.

Comportamiento

Una casilla de marca actúa como un conmutador. Cuando es utilizada por el usuario, su valor cambia de no marcado a marcado o de marcado a no marcado. El valor de las casillas de marca es independiente del valor de cualquier otro componente.

Funcionalidad interactiva:

- activación;
- conmutación entre los valores marcado y no marcado;
- validación.

Aspecto visual (véase la Figura 33)

Atributos

- **CIN:** Este atributo lleva el número de identificación de componente.
- **XPos:** Este atributo lleva la posición horizontal del elemento en NDC.
- **YPos:** Este atributo lleva la posición vertical del elemento en NDC.
- **Width:** Este atributo lleva la anchura del componente.
- **Height:** Este atributo lleva la altura del componente.
- **Closed:** El elemento estará en el estado cerrado.
- **NotAccessible:** El elemento no estará accesible.
- **TextLabel:** Este atributo lleva la etiqueta de texto del componente.
- **LabelFont:** El FIN de la etiqueta.
- **DefMarked:** Se marcará el valor por defecto del elemento.
- **LocActAct:** Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- **LocActVal:** Este atributo lleva el código de la acción local asociada con el componente y generada por su validación.

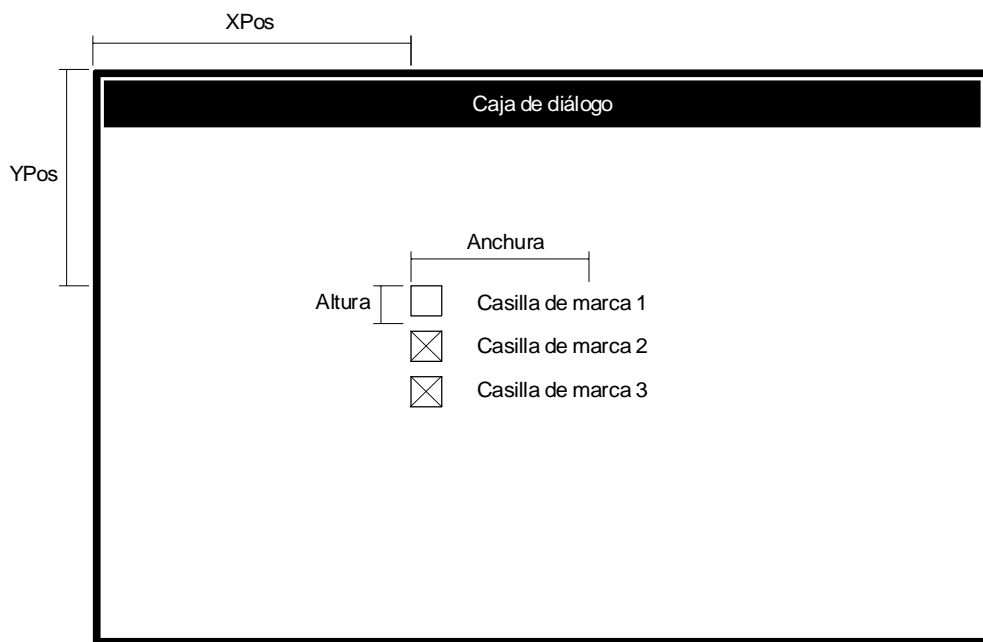


FIGURA 33/T.107
Casilla de marca

7.6.1.10 Componente botón radio

Descripción

El componente botón radio tiene dos ítems:

- el ítem etiqueta de texto;
- el ítem marca.

Para activar el componente puede usarse una clave mnemónica (véase 7.1.7)

El ítem de marca soportará dos opciones diferentes, visualmente identificables, que representan los dos posibles valores: marcado y no marcado. El espacio asignado al componente incluirá, en sus dimensiones, el espacio necesario para representar este ítem de marca.

Los botones radio suelen utilizarse en grupos para proporcionar un campo de una opción.

Comportamiento

No se marcará más de un botón radio en un grupo de botones radio en un momento dado. El intento de marcar un segundo botón radio suprimirá la marca del botón radio marcado anteriormente, independiente de su estado dentro de la aplicación (abierto, cerrado, accesible, inaccesible).

Funcionalidad interactiva

- activación;
- conmutación entre los valores marcado y no marcado;
- validación.

Aspecto visual (véase la Figura 34)

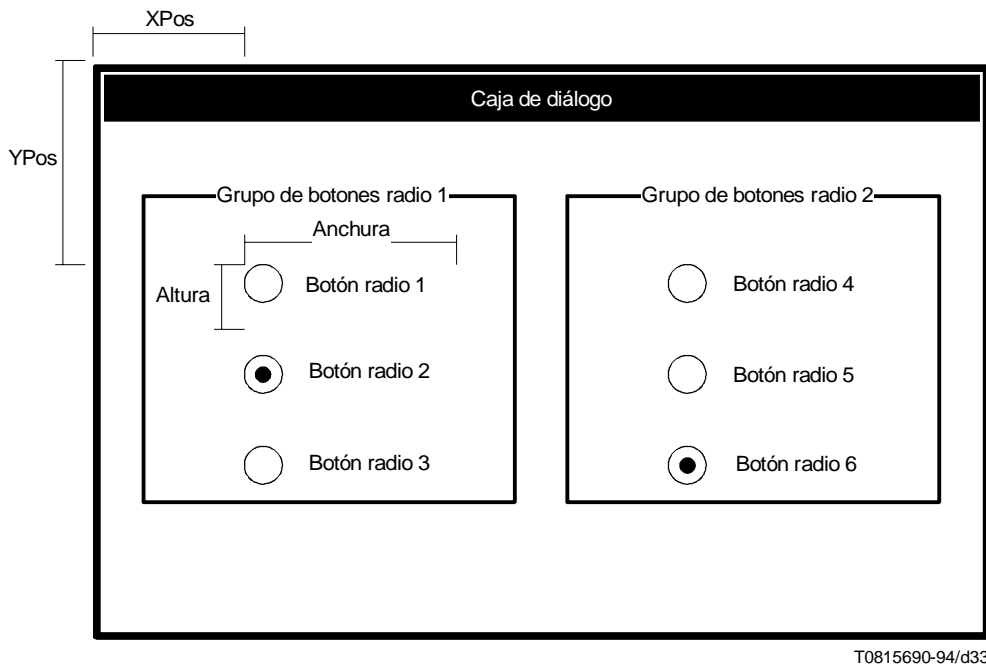


FIGURA 34/T.107

Botones radio con un marco y una zona de presentación de texto

Atributos

- CIN: Este atributo lleva el número de identificación de componente.
- Xpos: Este atributo lleva la posición horizontal del elemento en NDC.
- Ypos: Este atributo lleva la posición vertical del elemento en NDC.
- Width: Este atributo lleva la anchura del componente.
- Height: Este atributo lleva la altura del componente.
- Closed: El elemento estará en el estado cerrado.
- NotAccessible: El elemento no estará accesible.
- TextLabel: Este atributo lleva la etiqueta de texto del componente.
- LabelFont: El FIN de la etiqueta.
- Group: Este atributo lleva el identificador del grupo de botones radio. Sólo un botón radio del grupo podrá estar marcado en un momento dado.
- DefMarked: Se marcará el valor por defecto del elemento.
- LocActAct: Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- LocActVal: Este atributo lleva el código de la acción local asociada con el componente y generada por su validación.

7.6.1.11 Componente casilla de lista

Descripción

El componente casilla de lista es una zona rectangular en la cual se ofrece a selección del usuario una lista de ítems de texto. Si el número total de ítems rebasa el número de ítems presentados, se ofrecerán al usuario facilidades de desplazamiento vertical a través de controles de desplazamiento especializados. Cuando toda la lista es visible, los controles de desplazamiento vertical pueden ser invisibles. En este caso, el espacio asignado al componente incluirá el espacio necesario para representar las facilidades de desplazamiento. Las herramientas que proporcionan la funcionalidad de desplazamiento dependen del terminal (barra de desplazamiento vertical, botones, etc.).

Pueden añadirse pequeños iconos a los ítems de lista para realzar visualmente su significado. El mapa de bits referenciado puede redimensionarse para ajustarlo a la representación de casilla de lista local.

La lista completa se compone de ítems de lista. No habrá CR + LF dentro del contenido de texto de un ítem de lista. Cuando se selecciona el atributo clasificado, los ítems de lista se clasificarán por orden alfabético. Si no se selecciona la clasificación, la lista se presentará con el orden ascendente de los índices de ítem de lista. No obstante, se mantiene siempre la correspondencia entre los ítems de lista y los índices.

Comportamiento

Cuando la casilla de lista está activa, el terminal ofrecerá al usuario facilidades de cambio. Una casilla de lista contiene una lista de ítems que el usuario puede desplazar verticalmente y efectuar una selección. El usuario puede hacer una o múltiples selecciones, según los atributos de la lista.

Puede haber acciones locales asociadas a cada ítem de casilla de lista. Además, la casilla de lista también puede tener acciones locales asociadas. Al activar y validar un ítem de casilla de lista, se ejecutarán las acciones locales asociadas con el ítem de casilla de lista. Si no hay ninguna acción local asociada con el ítem seleccionado, se ejecutará la acción local asociada con la propia casilla de lista. En este caso, cualquier OIN que aparezca en las instrucciones específicas de elemento del componente casilla de lista se incrementará en el valor del índice de lista del ítem de casilla de lista.

Una casilla de lista de selección múltiple no puede validarse.

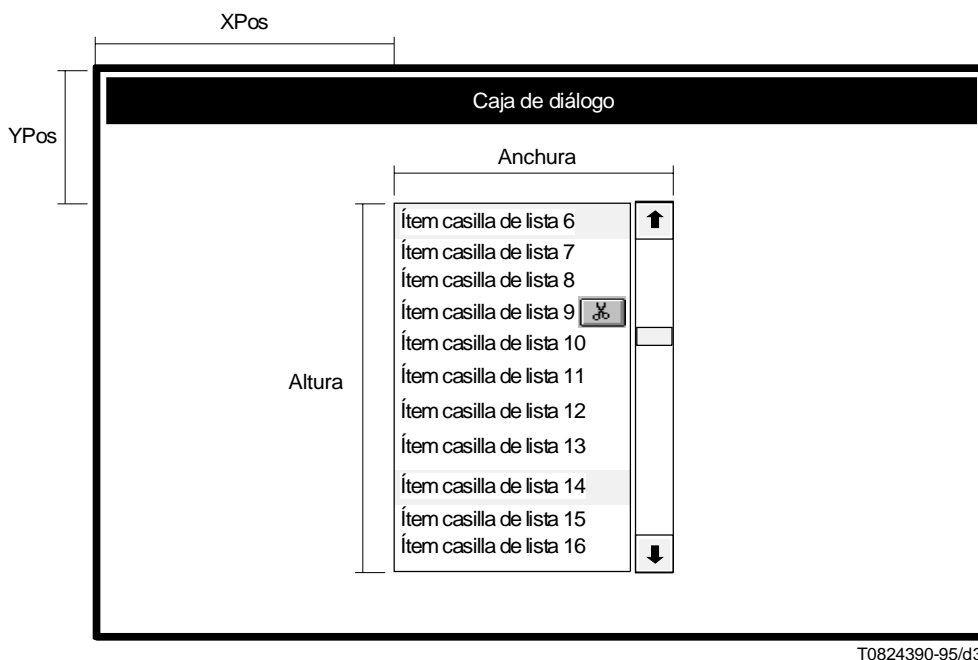
Funcionalidad interactiva

- activación;
- opción simple o múltiple;
- validación (una sola opción);
- desplazamiento vertical;
- cambio.

Aspecto visual (véase la Figura 35)

Atributos

- CIN: Este atributo lleva el número de identificación de componente.
- XPos: Este atributo lleva la posición horizontal del elemento en NDC.
- YPos: Este atributo lleva la posición vertical del elemento en NDC.
- Width: Este atributo lleva la anchura del componente.
- Height: Este atributo lleva la altura del componente.
- Closed: El elemento estará en el estado cerrado.
- NotAccessible: El elemento no estará accesible.
- DefItem: Este atributo lleva el índice del ítem de lista seleccionado por defecto cuando se abre el componente por primera vez.
- Sorted: Los ítems de la lista se presentan clasificados por orden alfabético.
- MultipleChoice: El elemento permite múltiples opciones.
- LocActAct: Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- LocActVal: Este atributo lleva el código de la acción local asociada con el componente y generada por su validación.



T0824390-95/d34

FIGURA 35/T.107

Casilla de lista

Atributos de ítem de lista

- **ListIndex:** Este atributo lleva el índice del ítem de lista siguiente en la descripción de componente. En una instrucción de modificación, posibilita la modificación de un ítem de lista específico. Los índices de lista se definirán en orden ascendente. Los índices de lista podrán no ser consecutivos para facilitar la actualización de la lista (inserción de ítems de lista). El terminal presentará al usuario una lista consecutiva, seleccionada por orden ascendente de los índices de lista correspondientes al texto de lista. Cuando en una instrucción de modificación se utiliza un índice de lista para un ítem de lista que ya existe, se reemplazará el ítem de lista existente. Cuando en una instrucción de modificación se utiliza un índice de lista para un ítem de lista que no existe en el terminal, se añadirá el nuevo ítem de lista. Cuando en una instrucción de modificación se referencia un índice de lista que ya existe, pero en esa instrucción de modificación falta el atributo «ListText», entonces se borrará del terminal el ítem de lista referenciado.
- **ListText:** Este atributo lleva el contenido de texto de un ítem de lista.
- **BIN:** El BIN del componente.
- **ItemActAct:** Este atributo lleva el código de la acción local asociada con el ítem y generada por su activación.
- **ItemActVal:** Este atributo lleva el código de la acción local asociada con el ítem y generada por su validación.

7.6.1.12 Componente casilla de combinación

Descripción

La casilla de combinación se compone de dos partes:

- una casilla de lista de una opción;
- un campo de texto de una línea situado en la parte superior de la lista.

Si el número total de ítems rebasa el número de ítems presentables, se ofrecerá al usuario la facilidad de desplazamiento vertical mediante facilidades de desplazamiento especializadas. En este caso, el espacio asignado al componente incluirá el espacio necesario para representar las facilidades de desplazamiento vertical. Deben reservarse dos caracteres para las herramientas de desplazamiento. Las herramientas que proporcionan la funcionalidad de desplazamiento dependen del terminal (barra de desplazamiento vertical, botones, etc.).

Pueden añadirse pequeños iconos a los ítems de lista para realzar visualmente su significado. El mapa de bits referenciado puede redimensionarse para ajustarlo a la representación de casilla de combinación local.

La lista completa se compone de ítems de lista. No habrá CR + LF dentro del contenido de texto de un ítem de lista. Cuando se selecciona el atributo clasificado, los ítems de lista se clasificarán por orden alfabético. Si no se selecciona la clasificación, la lista se presentará con el orden ascendente de los índices de ítem de lista. No obstante, se mantiene siempre la correspondencia entre los ítems de lista y los índices.

Una variación de la casilla de combinación es la casilla de combinación con exclusión (drop down combination box). Se compone de una casilla de combinación y un botón pulsado. Sólo se visualiza el campo de texto y el botón pulsado hasta que el usuario valida el botón pulsado asociado. La validación del botón pulsado origina la visualización de la casilla de lista asociada.

Comportamiento

Cuando la casilla de lista está activa, el terminal ofrecerá al usuario facilidades de cambio. Una casilla de combinación contiene una lista de ítems que el usuario puede desplazar verticalmente y seleccionarlos para completar el campo de texto. Un parámetro de la casilla de combinación especifica si el campo de texto puede editarse o no. Si el campo de texto se puede editar, el usuario puede mecanografiar texto directamente en el campo de texto. Un parámetro del componente indica si el texto mecanografiado concordará con uno de los ítems contenidos en la lista.

Puede haber acciones locales asociadas a cada ítem de casilla de lista. Además, la casilla de combinación también puede tener acciones locales asociadas. Al activar y validar un ítem de casilla de lista, se ejecutarán las acciones locales asociadas con el ítem de casilla de lista. Si no hay ninguna acción local asociada con el ítem seleccionado, se ejecutará la acción local asociada con la propia casilla de lista. En este caso, cualquier OIN que aparezca en las instrucciones específicas de elemento del componente casilla de lista se incrementará en el valor del índice de lista del ítem de casilla de lista.

Si la casilla de combinación está diseñada como una casilla de combinación con exclusión, la validación del botón pulsado conmutará entre la visualización y la ocultación de la casilla de lista.

Funcionalidad interactiva

- activación;
- selección simple;
- validación;
- desplazamiento vertical;
- edición.

Aspecto visual (véase la Figura 36)

Atributos

- CIN: Este atributo lleva el número de identificación de componente.
- XPos: Este atributo lleva la posición horizontal del elemento en NDC.
- YPos: Este atributo lleva la posición vertical del elemento en NDC.
- Width: Este atributo lleva la anchura del componente.
- Height: Este atributo lleva la altura del componente.
- Closed: El elemento estará en el estado cerrado.
- NotAccessible: El elemento no estará accesible.

- DefItem: Este atributo lleva el índice del ítem de lista seleccionado por defecto cuando se abre el componente por primera vez.
- NoEdit: El campo de texto del componente no será editable.
- NoConsistency: Este atributo se aplica si no está fijado «NoEdit». Indica que el texto introducido por el usuario puede ser cualquier cadena de texto, y no necesariamente uno de los ítems de texto definidos.
- MaxChar: Este atributo lleva el máximo número de caracteres que el usuario puede introducir en el componente. Sólo estará presente si el parámetro «EditableInput» está puesto en verdadero.
- NoDropDown: El elemento no será excluido.
- NotSorted: Los ítems de la lista no se presentan clasificados por orden alfabético.
- LocActAct: Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- LocActVal: Este atributo lleva el código de la acción local asociada con el componente y generada por su validación.

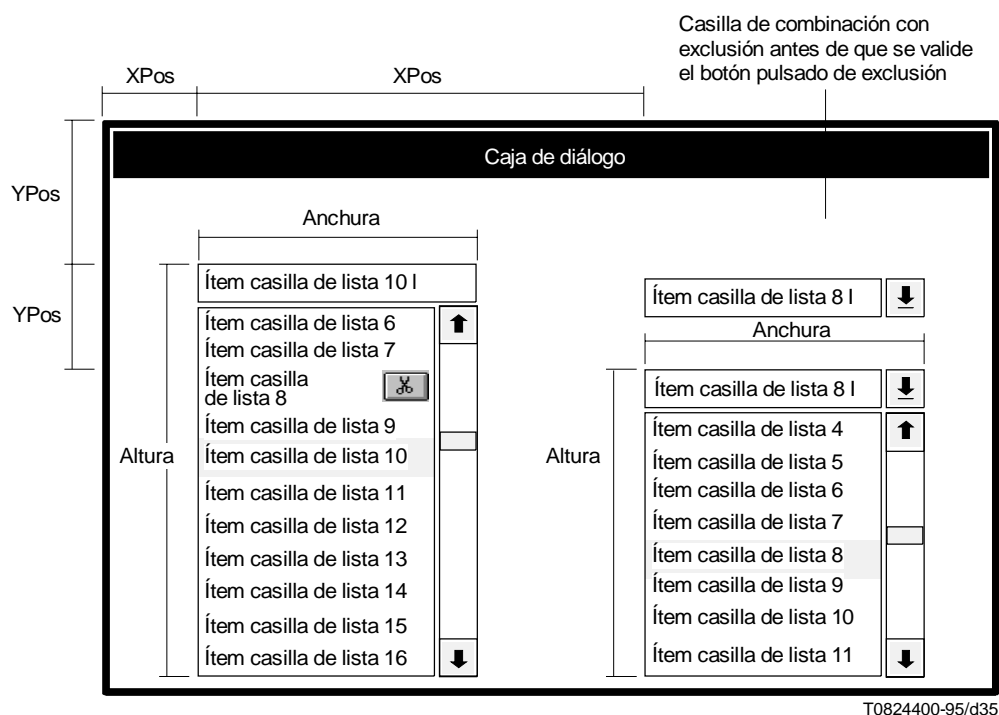


FIGURA 36/T.107
Casilla de combinación

Atributos de ítem de lista

- **ListIndex:** Este atributo lleva el índice del ítem de lista siguiente en la descripción de componente. En una instrucción de modificación, posibilita la modificación de un ítem de lista específico. Los índices de lista se definirán en orden ascendente. Los índices de lista podrán no ser consecutivos para facilitar la actualización de la lista (inserción de ítems de lista). El terminal presentará al usuario una lista consecutiva, seleccionada por orden ascendente de los índices de lista correspondientes al texto de lista. Cuando en una instrucción de modificación se utiliza un índice de lista para un ítem de lista que ya existe, se reemplazará el ítem de lista existente. Cuando en una instrucción de modificación se utiliza un índice de lista para un ítem de lista que no existe en el terminal, se añadirá el nuevo ítem de lista. Cuando en una instrucción de modificación se referencia un índice de lista que ya existe, pero en esa instrucción de modificación falta el atributo «ListText», entonces se borrará del terminal el ítem de lista referenciado.
- **ListText:** Este atributo lleva el contenido de texto de un ítem de lista.
- **BIN:** El BIN del componente.
- **ItemActAct:** Este atributo lleva el código de la acción local asociada con el ítem y generada por su activación.
- **ItemActVal:** Este atributo lleva el código de la acción local asociada con el ítem y generada por su validación.

7.6.1.13 Componente deslizador

Descripción

El deslizador ofrece la selección de un valor análogo al desplazar un marcador ajustable sobre una barra de desplazamiento entre un valor máximo y uno mínimo. Los intervalos los fija la aplicación.

El componente deslizador consta de los ítems siguientes:

- a) deslizador (con el marcador desplazable, la posición inicial será la del valor por defecto fijado por la aplicación);
- b) TextLabel (nombra al deslizador);
- c) MinLabel (valor seleccionable más bajo), opcional;
- d) MaxLabel (valor seleccionable más alto), opcional;
- e) DirectInputField (opcional), un campo de entrada y de salida, que contiene:
 - 1) el valor inicial al activar el deslizador;
 - 2) los valores en forma de texto correspondientes a los desplazamientos del deslizador;
 - 3) un campo de entrada para las entradas de usuario de la posición de deslizador.

El componente deslizador puede estar orientado en sentido vertical u horizontal. Todos los ítems cabrán en el rectángulo determinado por las dimensiones del componente.

Una acción local generada por la validación se ejecutará cuando el usuario desactiva el componente.

Comportamiento

Cuando el componente deslizador está activo, puede desplazarse el marcador pulsando las teclas de cursor o mediante clicks del ratón. Si el DirectInputField está presente, el usuario podrá introducir valores directamente mediante las teclas. Los valores modificados se visualizarán de inmediato en el DirectInputField. El marcador se desplazará a una posición correspondiente a esa entrada de valor. En el otro caso, cuando se desliza el marcador, el valor visualizado se referirá a este desplazamiento. MinLabel y MaxLabel (si están presentes) indican la gama dentro de la cual se podrá hacer una selección.

Funcionalidad interactiva

- funcionalidad de entrada local de DirectInputField (depende de la aplicación);
- posicionamiento local del marcador deslizador;
- validación por desactivación del componente;
- validación.

Aspecto visual (véase la Figura 37)

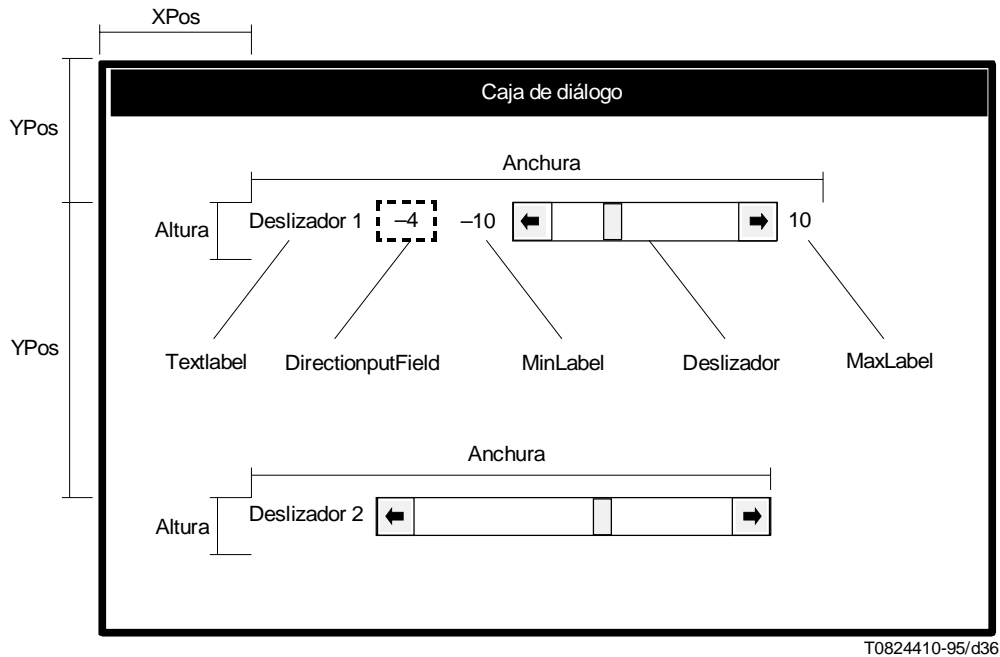


FIGURA 37/T.107

Componente deslizador

Atributos

- CIN: Este atributo lleva el número de identificación de componente.
- XPos: Este atributo lleva la posición horizontal del elemento en NDC.
- YPos: Este atributo lleva la posición vertical del elemento en NDC.
- Width: Este atributo lleva la anchura del componente.
- Height: Este atributo lleva la altura del componente.
- Closed: El elemento estará en el estado cerrado.
- NotAccessible: El elemento no estará accesible.
- Label: Este atributo lleva una cadena de texto alfanumérico.
- LabelFont: El FIN de la etiqueta.

- **MinValue:** El valor ajustable más bajo.
- **MaxValue:** El valor ajustable más alto.
- **Increment:** Tamaño del intervalo entre dos valores ajustables.
- **InitialValue:** El valor inicial, cuando se abre el componente deslizador.
- **Negative:** Este atributo se puede asociar con **MinValue**, **MaxValue** e **InitialValue**, e indica que los valores son negativos.
- **DirectIn:** Este atributo indica que **MinValue** y **MaxValue** serán visualizados y que se permitirán las entradas directas del usuario.
- **Vertical:** El elemento se presentará vertical.
- **LocActAct:** Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- **LocActVal:** Este atributo lleva el código de la acción local asociada con el componente y generada por su validación.

7.6.1.14 Componente zona sensible

Descripción

Un componente zona sensible es una zona, no materializada, en la que la aplicación permite una operación de validación. Se prevé que el componente zona sensible se utilizará en asociación con un componente zona de salida gráfica de la casilla de diálogo. Para el uso con componentes de texto, los atributos in-text resultan más apropiados para la definición del texto sensible. La zona sensible es un rectángulo dentro de la casilla de diálogo.

Si se fija el localizador de atributo, el terminal enviará tras la validación no sólo su CIN a la aplicación VEMMI, sino también las coordenadas del cursor en NDC en el momento de la validación con respecto al origen de componentes.

Comportamiento

Cuando está activa, puede implementarse un efecto visible (por ejemplo, líneas de rayas o de puntos).

Constricciones

Una zona sensible no será cubierta por otros componentes o ítems que son sensibles a la interacción de usuario, para evitar conflictos.

Funcionalidades interactivas

- activación;
- validación.

Aspecto visual (véase la Figura 38)

Atributos

- **CIN:** Este atributo lleva el número de identificación de componente.
- **XPos:** Este atributo lleva la posición horizontal de la zona sensible.
- **YPos:** Este atributo lleva la posición vertical de la zona sensible.
- **Width:** Este atributo lleva la anchura de la zona sensible.
- **Height:** Este atributo lleva la altura de la zona sensible.
- **Closed:** El elemento estará en el estado cerrado.
- **NotAccessible:** El elemento no estará accesible.

- Localizador: La posición del cursor en NDC en el momento de la validación con respecto al origen de componentes se enviará con cada instrucción de informe.
- LocActAct: Este atributo lleva el código de la acción local asociada con el componente y generada por su activación.
- LocActVal: Este atributo lleva el código de la acción local asociada con el componente y generada por su validación. Si el atributo localizador está fijado, cada acción de informe contendrá también las coordenadas del cursor en NDC en el momento de la validación con respecto al origen de componentes.

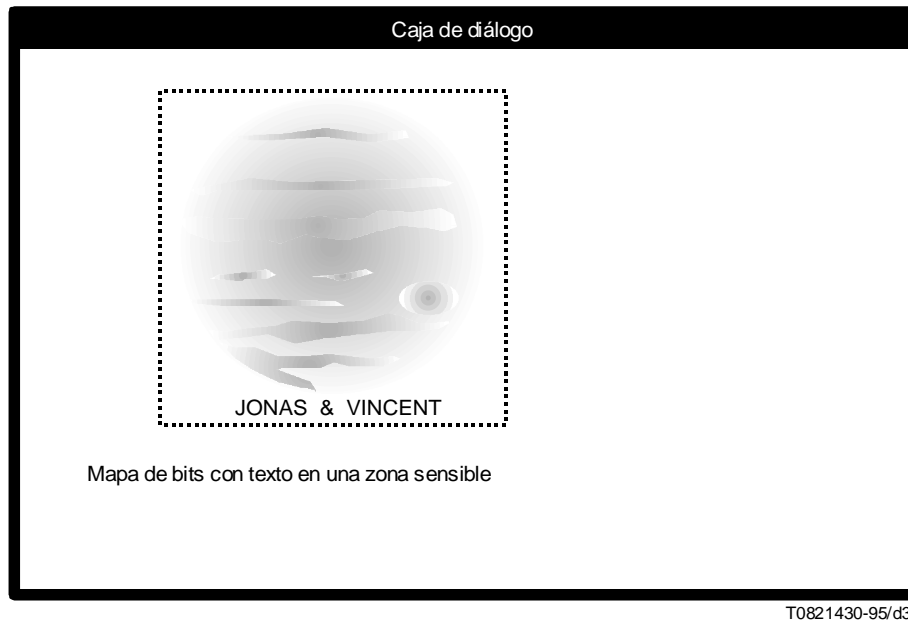


FIGURA 38/T.107
Mapa de bits, texto y zona sensible

7.7 Casilla de mensaje

Descripción

La casilla de mensaje es una zona rectangular en la DDA que contiene información de texto.

Se definen cuatro tipos especificados de mensajes:

- mensaje general;
- mensaje de información;
- mensaje de aviso;
- mensaje de acción.

Esta información puede utilizarse para añadir un icono que caracteriza el tipo del mensaje en la presentación de la casilla de mensaje.

Un atributo de la casilla de mensaje especifica si el terminal debe producir un sonido cuando se abre dicha casilla.

La casilla de mensaje puede tener un límite que debe ser igual a la posición de un carácter. Cuando se solicita un límite, el terminal dibujará un marco. Este dibujo del marco depende del terminal.

Comportamiento

Un atributo de casilla de mensaje especifica la duración de esa casilla. Se dispone de los cinco valores siguientes:

- destrucción por cualquier interacción de usuario;
- cierre por cualquier interacción de usuario;
- destrucción por la validación de usuario de un botón definido implícitamente;
- cierre por la validación de usuario de un botón definido implícitamente;
- no se define ninguna duración implícita.

La destrucción o el cierre de la casilla de mensaje no motivará ningún mensaje al servidor.

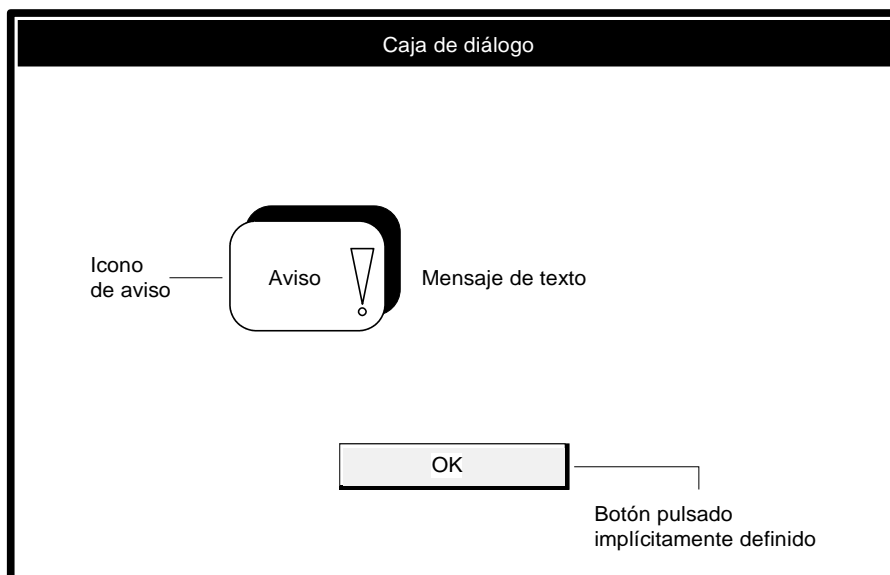
Si la aplicación VEMMI no define ninguna posición, el terminal centrará la casilla de mensaje en la DDA.

Si la aplicación VEMMI no define dimensiones, el terminal calculará el tamaño apropiado de la casilla de mensaje, con el fin de tener espacio suficiente para visualizar el mensaje de texto y el posible botón pulsado. La etiqueta y el tamaño del posible botón pulsado dependen del terminal.

Funcionalidad interactiva

- desplazamiento;
- validación.

Aspecto visual (véase la Figura 39)



T0815790-94/d38

FIGURA 39/T.107

Casilla de mensaje con un mensaje de aviso

Atributos

- XPos: Este atributo lleva la posición horizontal del elemento en NDC.
- YPos: Este atributo lleva la posición vertical del elemento en NDC.
- Width: Este atributo lleva la anchura del componente.
- Height: Este atributo lleva la altura del componente.
- Closed: El elemento estará en el estado cerrado.
- MessageType: Uno de: general, información, aviso, acción.
- Modal: El elemento será modal.
- NoBorder: El elemento no tendrá un límite.
- AttributedText: Texto del mensaje con los atributos in-text.
- MaxTime: Periodo de tiempo expresado en segundos.
- Lifetime: Los eventos que llevan a la destrucción de la casilla de mensaje.
- NoSound: El terminal no debe ejecutar un sonido cuando se abre la casilla de mensaje.

7.8 Objeto operativo

Con este objeto una aplicación referencia un programa que se enlazará con la aplicación VEMMI. Este tipo de objeto proporciona un método para ampliar las capacidades de una aplicación durante el tiempo de ejecución. Por medio de VEMMI_Crear_objeto, la aplicación anfitriona define el objeto operativo. El programa se arranca con VEMMI_Abrir_objeto. Una vez terminado, se devuelve el control a la aplicación.

Ese programa se ha cargado mediante una transferencia de ficheros, y es parte del sistema operativo o su presencia se ha negociado de alguna otra manera. Se referencia mediante su nombre de fichero. Le corresponde al anfitrión, que suministrará los parámetros de programa correctos, garantizar que el programa llamado arranca sin fallos. Una técnica que se utiliza con frecuencia es ampliar las funciones de los programas durante el tiempo de ejecución mediante el empleo de bibliotecas de enlace dinámico, DLL.

El programa referenciado con un objeto operativo puede ser del tipo siguiente:

- programa independiente;
- programa con una interfaz de filtro.

Para las finalidades de los objetos operativos se usan los servicios VEMMI_Crear_objeto, VEMMI_Abrir_objeto y VEMMI_Destruir_objeto. El nombre del programa y sus parámetros se definen como atributos del objeto operativo.

El terminal puede emplear dos parámetros: descripción de programa e identificación de fuente, para evaluar la información que se le presentará al usuario o que sólo se grabará. Es necesario tomar esta precaución para ayudar al usuario a detectar los programas que se puedan haber enlazado a la aplicación VEMMI con fines de sabotaje. Su efecto en el terminal puede acarrear daños. El terminal puede comunicar al anfitrión: «El usuario se niega a arrancar el programa». Los motivos para generar esta respuesta dependen de la aplicación, pero al menos se dispone de la información sobre una base estructurada. Si el terminal no puede encontrar el programa, informará al anfitrión mediante un mensaje de error específico.

Requisitos de los programas independientes

El programa no tiene un canal de intercambio de datos con el gestor local VEMMI durante la ejecución. La aplicación anfitriona puede definir los parámetros de programa y proporcionarlos al programa cuando éste arranca. El objeto operativo no tiene efectos visuales en el terminal, y el usuario no puede activarlo directamente, pero el programa iniciado puede utilizar el UI local sobre una base específica de aplicación. Véase la Figura 40.

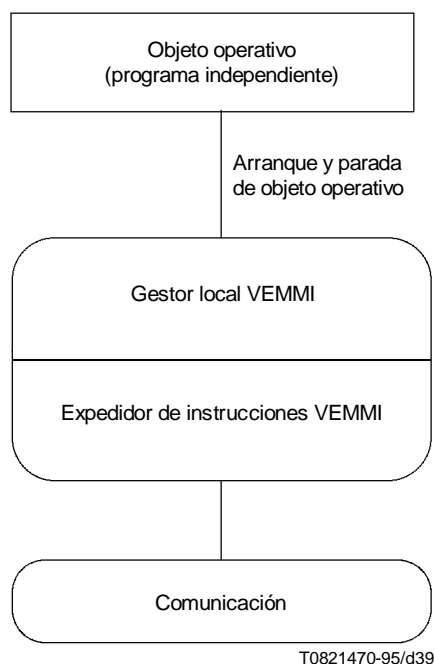


FIGURA 40/T.107
Programas independientes

El expedidor de instrucciones VEMMI se encarga de la entrega de las instrucciones VEMMI al gestor local VEMMI y a la parte de comunicación. Todas estas interfaces dependen del terminal y de la aplicación. Se utilizan en esta descripción para explicar la integración de los dos tipos de objeto operativo de la arquitectura de terminal VEMMI.

Requisitos de los programas con interfaz de filtro

El programa con interfaz de filtro tiene la posibilidad de acceder al expedidor de instrucciones VEMMI. El expedidor de instrucciones envía instrucciones al objeto operativo, que decide entre procesarlas o enviarlas al gestor local VEMMI. Este mecanismo funciona también en el sentido remitente. El objeto operativo no tiene efectos visuales en el terminal, y el usuario no puede activarlo directamente, pero el programa iniciado puede utilizar el UI local sobre una base específica de aplicación o el gestor local VEMMI. Todas estas interfaces dependen del terminal y de la aplicación, y no se las describe en esta Recomendación. Cuando un objeto operativo recibe una instrucción VEMMI_Cerrar_objeto con su propio OIN procedente del anfitrión, o es terminado por el gestor local VEMMI, se detendrá, y el gestor local VEMMI continuará. Se desactivan todos los canales de intercambio de datos con el gestor local VEMMI. Véase la Figura 41.

Atributos

- Closed: El elemento estará en el estado cerrado.
- ProgName: El nombre del programa.
- ProgFilename: Nombre de fichero del programa.
- ProgDescr: Descripción del programa.
- ProgAbout: Identificación de fuente del programa.
- ProgPar: Lista de parámetros para el arranque del programa (opcional).
- ProgType: Tipo programa, autónomo o con interfaz de filtro.

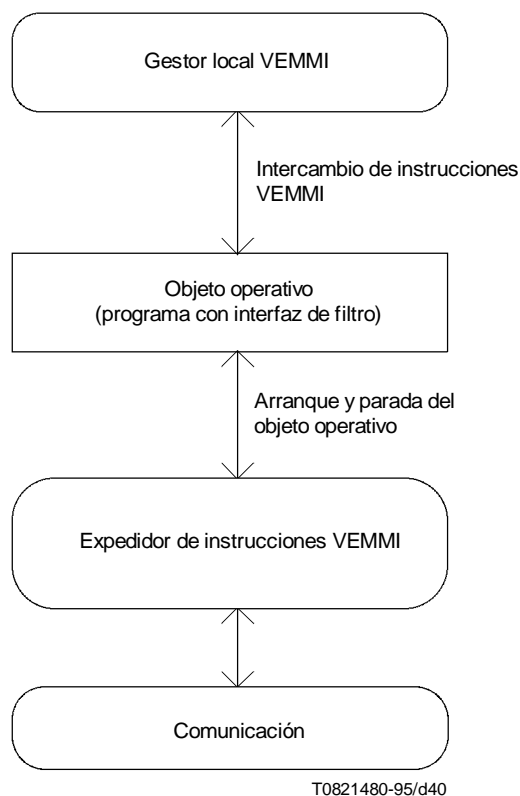


FIGURA 41/T.107
Programas con interfaces de filtro

7.9 Objeto recurso mapa de bits

Descripción

Un objeto mapa de bits contiene la definición de mapa de bits propiamente dicha o sólo una referencia a un fichero con la definición de mapa de bits.

Definición de mapa de bits directa

La matriz de pixels contiene índices de la tabla de colores o el componente RGB de cada pixel. La lista de atributos consta de dos partes: la primera es común a los dos casos, y la segunda depende de la definición de color. La primera parte consiste en:

- BmWidth: Anchura del mapa de bits expresada en pixels.
- BmHeight: Altura del mapa de bits expresada en pixels.
- BmCompr: Tipo de compresión de la lista de colores. Este valor no se define actualmente, y sólo se pueden especificar listas de colores no comprimidas (opcionales).

En el caso de la definición de índice de color, la segunda parte de la lista de atributos consiste en:

- BmBitsPerPixel: Número de bits por pixel. El valor será 1, 4 u 8 (valor por defecto opcional: 1).
- BmClrEntry: El índice de la tabla de colores en el que se basan todos los demás índices de color de la lista (valor por defecto opcional: 0).

- **BmClrIxList:** Lista de índices de color. La lista contiene índices de color de $BmWidth * BmHeight$. Según el número de bits por pixel, cada índice de color tiene una gama $0 \dots 2^{BmBitsPerPixel} - 1$. Los índices de color se ordenan fila por fila, de izquierda a derecha. El punto de comienzo es la esquina inferior izquierda del mapa de bits. Un índice de color 0 significa un índice de tabla de colores igual a **BmClrEntry**. Todos los demás índices se basan en **BmClrEntry**.

En el caso de la definición del componente Color, la segunda parte de la lista de parámetros consiste en:

- **BmBitsPerComp:** Número de bits por componente color. El valor debe estar dentro de la gama 1-8 (valor por defecto opcional: 8).
- **BmClrCompList:** Lista de tripletes RGB. La lista contiene tripletes de $BmHeight * BmWidth$. Los tripletes se ordenan fila por fila, de izquierda a derecha. El punto de comienzo es la esquina inferior izquierda del mapa de bits.

Definición de mapa de bits de fichero

Este tipo de mapa de bits se define a partir de un fichero de imagen almacenado en el terminal. Ese fichero fue cargado mediante un servicio de transferencia de recursos o por algún otro medio. Un parámetro indica la codificación del fichero. Si no se pudo convertir el fichero a un mapa de bits adecuado (con **VEMMI_Crear_objeto**), el terminal enviará un mensaje de error al anfitrión.

- **Filename:** Nombre del fichero que contiene la definición del mapa de bits.
- **PictFileType:** Tipo de fichero imagen (uno de JPEG, GIF o BMP).

Los ficheros imagen se conforman a una de las especificaciones siguientes:

- **JPEG:** Sistema de referencia de ISO/CEI 10918 | Rec. UIT-T T.81 «Compresión digital y codificación de imágenes de tonos continuos».
- **GIF:** Formato de intercambio de gráficos (sm) versión 89a. CompuServe Incorporated, Columbus, Ohio, EE.UU. A pesar de que la versión 89a está referenciada, el terminal tiene que soportar sólo las funciones de la versión 87.
- **BMP:** Mapa de bits de Microsoft Windows independiente del dispositivo (device-independent mapa de bits, DIB), con RLE4, RLE8 y 1, 4, 8 ó 24 bits por pixel.

7.10 Objeto de recurso videotex

Descripción

Un objeto videotex contiene el contenido videotex propiamente dicho o sólo una referencia a un fichero que tiene el contenido videotex.

Atributos

- **VTX:** Contenido VTX.
- **Filename:** Nombre del fichero que tiene el contenido videotex.

7.11 Objeto recurso texto

Descripción

Este objeto define el contenido de texto como un recurso que puede ser referenciado mediante el TIN (número de identificación de texto). Contiene el contenido de texto propiamente dicho o una referencia a un fichero que tiene el contenido de texto. En ambos casos, el contenido de texto puede llevar atributos in-text, con referencias a objetos recurso tipo de caracteres.

Atributos – Definición de recurso texto directa

- **Text:** Texto.
- **FIN:** Referencia a un objeto recurso tipo de caracteres.

Atributos – Definición de recurso texto de fichero

- **Filename:** Nombre del fichero que tiene el contenido de texto.
- **TextFileType:** Una de:
 - texto;
 - referencias a objetos recurso tipo de caracteres y texto.

7.12 Objeto recurso tipo de caracteres

Este objeto combina un conjunto de atributos de recurso tipo de caracteres, que pueden ser referenciados mediante FIN.

Atributos

- **FnFamily:** Familia de tipos de caracteres: SWISS, ROMAN o FIXFONT (valor por defecto opcional: SWISS).
- **FnHeight:** Altura del carácter (valor por defecto opcional: 10).
- **FnBold:** Negrura de los caracteres en negrita (opcional).
- **FnUnderline:** Subrayado (opcional).
- **FnItalic:** Cursiva (opcional).
- **FnColour:** Color de los caracteres: índice de color (valor por defecto opcional: negro).

7.13 Objeto metacódigo

El objeto metacódigo contiene instrucciones VEMMI. Este objeto ofrece un medio fácil para evitar pasos de diálogo innecesarios con la aplicación anfitriona. La instrucción VEMMI_Abrir_objeto comienza el procesamiento del objeto metacódigo que, después de terminado, pasa automáticamente al estado cerrado. Las instrucciones (el contenido del objeto metacódigo) se procesan igual que si se recibieran de la aplicación anfitriona. El contenido del objeto metacódigo está sujeto a los condicionantes siguientes:

- la creación de un objeto metacódigo no está permitida;
- sólo se permiten instrucciones enviadas por la aplicación anfitriona;
- no se permite una instrucción VEMMI_Abrir_objeto de otro objeto metacódigo (VEMMI_Abrir_objeto_bloqueante está autorizada).

Atributos

- **VEMMICommands:** Instrucciones VEMMI.

7.14 Definición de tipo de datos de mapa de bits VEMMI

Véase 7.9.

7.15 Catálogo de identificación de codificación de contenidos VEMMI

El Cuadro 38 ofrece una sinopsis de las identificaciones de codificación de contenidos definidas en el catálogo VEMMI.

CUADRO 38/T.107

Catálogo de identificación de codificación de contenidos VEMMI

Para la codificación de datos de texto:	
	Cadena definida en el Anexo A/T.51
	Recomendación T.52 [3]
	Texto VEMMI de alta calidad
	ISO 8859 [13] (caracteres acentuados bien formados)
	ISO 10646-1 [15] (juego de caracteres multiocteto) (unicódigo)
	Código JIS (caracteres del japonés)
Para la codificación de datos de imagen fija:	
	Anexo F/T.101 [4] (fotovideotex)
	ISO 10918-1 [16] (JPEG)
	Mapa de bits VEMMI independiente del dispositivo
	Formato de intercambio de gráficos (sm) versión 89a. CompuServe Incorporated, Columbus, Ohio, EE.UU.
	Mapa de bits Microsoft Windows independiente del dispositivo (DIB) con RLE4, RLE8 y 1, 4, 8 ó 24 bits por pixel
Para la codificación de datos gráficos:	
	Anexo B/T.101 y Anexo C/T.101 [4] (videotex)
	ISO 8632 [12] (CGM)
Para la codificación de datos audio:	
	Anexo E/T.101 [4]
	Formato WAVE
	Formato MIDI
Para la codificación de datos de imagen en movimiento:	
	ISO 11172-2 [18] (vídeo MPEG)
	Recomendación H.261 [7] (videofonía)
Para la codificación de datos audiovisuales:	
	ISO 11172-1 [17] (sistema MPEG)
	Recomendación H.320 [8] (videofonía)

8 Representación codificada completa de VEMMI

8.1 Introducción

Esta cláusula contiene la notación de sintaxis, el encabezamiento VEMMI y la sintaxis de las instrucciones, los objetos, los componentes y las acciones locales.

8.2 Notación utilizada

La notación a/b con a, b = 0..15 designa un valor de byte en hexadecimal.

Los bits de un byte se escriben b₈, b₇, ..., b₁. El bit b₈ es el de orden superior o bit más significativo.

La sintaxis de las instrucciones, los objetos, los componentes y las acciones locales se definen mediante una gramática formal en la que se utiliza la notación siguiente:

<symbol>	– no terminal;
<SYMBOL>	– terminal;
<a/b>	– terminal, valor hexadecimal: a, b = 0..15;
<symbol>*	– 0 o más ocurrencias;
<symbol>+	– 1 o más ocurrencias;
<symbol>o	– opcional (0 ó 1 ocurrencias);
<symbol-1> := <symbol-2>	– el símbolo-1 tiene la sintaxis del símbolo-2;
<symbol-1> <symbol-2>	– el símbolo-1 o alternativamente el símbolo-2;
<symbol-1 : symbol-2>	– símbolo con el valor declarado;
{ comment }	– explicación de un símbolo o producción.

8.3 Conmutación global del entorno de codificación

ISO/CEI 9281 [14] describe una técnica para identificar métodos de codificación. El modo VEMMI para videotex es uno de los métodos de codificación identificados por dicha Norma. El diagrama de la Figura 42 muestra una visión general de la relación entre la sintaxis de datos videotex y los entornos de codificación de ISO/CEI 9281 [14].

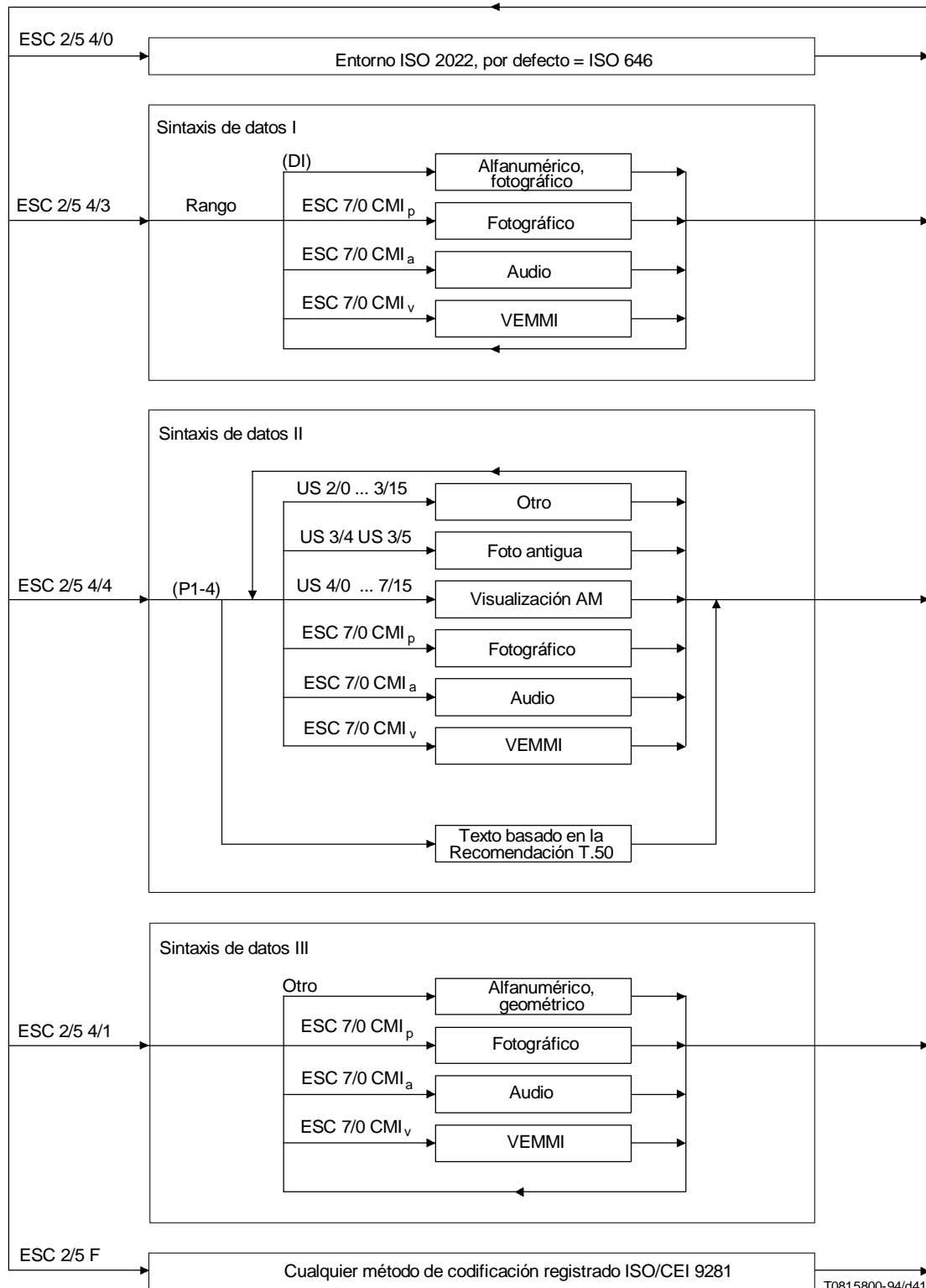
A partir de un entorno de ISO 2022 [10], puede introducirse explícitamente una sintaxis de datos videotex utilizando un código ESC 2/5 F. Éste es también el mecanismo utilizado para pasar de un entorno ISO 2022 [10] a un entorno ISO/CEI 9281 [14]. El código F («byte final») ha sido asignado y registrado, de acuerdo con ISO 2022 [10] por la autoridad de registro ISO 2375 [11]. De acuerdo con Apéndice B/ISO 2375 [11], las sintaxis de datos videotex se consideran como «sistemas de codificación diferentes al de ISO 2022 [10]». Los códigos F son 4/3 para la sintaxis de datos I del CCITT, 4/4 para la sintaxis de datos II del CCITT y 4/1 para la sintaxis de datos III del CCITT.

Como un terminal videotex suele comenzar el funcionamiento, por defecto, en una de las sintaxis de datos, no será obligatorio enviar primero un código ESC 2/5 F (F es 4/1, 4/3 ó 4/4). El diagrama muestra cómo pueden utilizarse estos códigos para conmutar un terminal videotex que soporta más de una sintaxis de datos de una sintaxis de datos a otra.

8.3.1 Conmutación al modo VEMMI

Un terminal videotex que funciona con una de las sintaxis de datos (es decir, un sistema de codificación distinto al descrito en ISO 2022 [10]) puede pasar al entorno de la ISO/CEI 9281 [14] del modo VEMMI de acuerdo con sus propias reglas. En el caso de videotex, para conmutar del entorno ISO/CEI 9281 [14] del modo VEMMI se utiliza el delimitador de código de imagen (PCD, *picture code delimiter*) del primer elemento de imagen (PE, *picture element*). El identificador de método de codificación (CMI, *coding method identifier*) se utiliza para distinguir entre métodos de codificación de imágenes. En el caso de videotex, éste será, por ejemplo, una distinción entre datos de audio, fotográficos y VEMMI.

NOTA – Como ISO/CEI 9281 [14] se elaboró especialmente para la identificación de codificación de imágenes en los entornos ISO 2022 [10], la palabra «imagen» se utiliza a menudo en las definiciones, incluso cuando es aplicable, por ejemplo, a «audio». La ISO ha aceptado ya utilizar ISO/CEI 9281 [14] para información distinta de imagen.



T0815800-94/d41

- DI es específico de la sintaxis de datos I
- P es un perfil en la sintaxis de datos II
- F es un código final asignado por la autoridad de registro de ISO 2022 [10]
- CMI_a es cualquier CMI (identificador de método de codificación) para datos audio videotex
- CMI_p es cualquier CMI para datos fotográficos videotex
- CMI_v es cualquier CMI para datos VEMMI
- Rango es específico de la sintaxis de datos I

FIGURA 42/T.107
Mecanismo de conmutación global

8.3.2 Estructura de la sintaxis ISO/CEI 9281 [14]

El formato de alto nivel de la sintaxis es el definido en ISO/CEI 9281 [14].

En la siguiente descripción de la sintaxis, se supone la codificación de 8 bits, por lo que se utiliza la palabra «byte» con el bit 8 puesto a cero. El método descrito en ISO 9281 [14] es también válido en un entorno de 7 bits. En este caso, se interpretará que la palabra «byte» significa «byte de 7 bits» y no se utilizará el bit más significativo, el bit 8.

La estructura de la codificación es la siguiente:

```

PE      ::= PCE PDE;
PCE     ::= PCD CMI LI;
PCD     ::= 01/11 07/00;
CMI     ::= PM PI;
PM      ::= 02/05 (modo VEMMI videotex);
PI      ::= 04/00;
LI1)  ::= x111 1111<byte1><byte2>...<byten>;
<bytek> ::= x10D DDDD (k = n);
          | x11D DDDD (1 = <k<n).
    
```

x indica «no importa».

D indica el número binario 0 ó 1.

Cada pieza de información, en este caso, datos VEMMI codificados, se codifica como una o más entidades de imagen (PE). Una PE (véase la Figura 43) consiste en una entidad de control de imagen (PCE) seguida de datos reales empaquetados en una entidad de datos de imagen (PDE).

Entidad de imagen (PE)			
Entidad de control de imagen (PCE)			PDE
PCD	Identificador de método de codificación		Entidad de datos de imagen
Delimitador de codificación de imagen	LI		
	Indicador de longitud		
	PM	PI	
	Modo de imagen	Identificador de imagen	

FIGURA 43/T.107

Estructura de una entidad de imagen

¹⁾ Para la descripción del indicador de longitud, véase 5.2.7/ISO 9281 [14].

NOTA 1 – En el modo VEMMI videotex, el tamaño de un fichero que contiene un objeto VEMMI codificado puede ser bastante grande. Se puede transmitir un objeto en varias PE. La utilización de varias PE podrá facilitar la terminación de la transmisión de un objeto por el usuario.

La entidad de control de imagen (PCE) consiste en un delimitador de codificación de imagen (PCD) y un identificador de método de codificación (CMI) seguido de un indicador de longitud (LI).

El delimitador de codificación de imagen (PCD) es una secuencia fija de dos octetos: 01/11 07/00.

NOTA 2 – 01/11 es ESC.

El identificador de método de codificación (CMI) consiste en un octeto de modo imagen (PM) seguido de un octeto de identificador de imagen (PI). Para VEMMI, el PM es 02/05 registrado por ISO 9281 [14]. El octeto PI tiene el valor 04/00.

El indicador de longitud (LI) especifica el número de bytes de la entidad de datos de imagen. Su codificación se describe a continuación (tomada de ISO/CEI 9281 [14]):

b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
X	UNO	CERO o UNO					

El bit 8 de cada byte se ignorará, el bit 7 de cada byte se pondrá a UNO. El bit 6 de cada byte será la bandera de extensión. El valor de LI se especifica en notación binaria como un número sin signo utilizando los bits 5 a 1 con las ponderaciones 2⁴, 2³, 2², 2¹ y 2⁰, respectivamente. Si el valor de LI es menor o igual que 31, se representará mediante un byte, y la bandera de extensión se pondrá a CERO. Si el valor de LI es mayor que 31, se representará mediante más de un byte. La parte más significativa de este valor estará contenida en el byte registrado primero. La bandera de extensión se pondrá a UNO en todos los bytes, salvo el último, donde se pondrá a CERO.

Valores de LI:

- 1 byte: X111 1111
- 2 bytes: $0 \leq LI \leq 2^5 - 1$
- 3 bytes: $2^5 \leq LI \leq 2^{10} - 1$
- \vdots
- \vdots
- n bytes: $2^{5(n-1)} \leq LI \leq 2^{5n} - 1$

Ejemplos:

LI = 31

X	1	0	1	1	1	1	1
---	---	---	---	---	---	---	---

LI = 33

X	1	1	0	0	0	0	1
---	---	---	---	---	---	---	---

Primer byte

X	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Segundo byte

8.4 Sintaxis de las instrucciones VEMMI

Instrucciones de anfitrión	Sintaxis de la instrucción
VEMMI_Fijar_modo_traducción	<translation mode opc> <translation mode>
VEMMI_Abrir	<open opc> <setup entity>
VEMMI_Cerrar	<close opc>
VEMMI_Reanudar	<resume opc> <mode>
VEMMI_Suspender	<suspend opc>
VEMMI_Cerrar_todos	<close all opc>
VEMMI_Identificar_cap_term	<term cap request opc> <cap request>
VEMMI_Fijar_opciones	<set options opc> <text type list>o
VEMMI_Restablecer_tabla_col	<reset col opc>
VEMMI_Bloquear_usuario	<user lock opc>
VEMMI_Desbloquear_usuario	<user unlock opc>
VEMMI_Abrir_objeto	<open object opc> <oin spec>
VEMMI_Cerrar_objeto	<close object> <oin spec>
VEMMI_Destruir_objeto	<destroy object opc> <oin spec>
VEMMI_Inhabilitar_acceso_objeto	<access disable opc> <oin spec>
VEMMI_Habilitar_acceso_objeto	<access enable opc> <oin spec>
VEMMI_Crear_objeto	<create object opc> <entity create object>
VEMMI_Borrar_objetos_antiguos	<delete outdated objects opc> <entity versionlist>+
VEMMI_Modificar_componente (véase también 9.6)	<modify comp opc> <entity modify comp>
VEMMI_Cambiar_ubicación_objeto	<object loc change opc> <oin> <xloc> <yloc>
VEMMI_Cargar_tabla_col	<load coltable opc> <entity load coltable>
VEMMI_Abrir_aplicación	<open appl opc> <entity open appl>
VEMMI_Almacenar_objetos	<store objects opc> <oin spec>
VEMMI_Borrar_objetos	<erase objects opc> <oin spec>
VEMMI_Abrir_objeto_bloqueante	<open block object opc> <oin>
VEMMI_Transferencia_recursos	<transfer opc> <transfer entity>
VEMMI_Resp_almacenar_objetos	<store objects resp opc> <store result>
VEMMI_Retransmisión_objeto	<object retrans opc> <oin>
VEMMI_Resp_identificar_cap_term	<term cap response opc> <cap response>+
VEMMI_Resp_abrir_aplicación	<open appl resp opc> <entity open appl resp>
VEMMI_Datos_usuario	<user data opc> <entity user data>
VEMMI_Error	<error opc> <entity errorreport>
VEMMI_Aborto_transferencia_recursos	<transfer opc> <transfer id>

<translation mode>:=	<2/0> { no translation } <2/1> { 3 in 4 encoding } <2/2> { 7 shift encoding }
<setup entity>:=	<version> <mode>
<version>:=	<INTEGER> { Version }
<mode>:=	<native> <page based>
<native>:=	<INTEGER : 0> { VEMMI as described in this Recommendation }
<page based>:=	<INTEGER : 1> <basic page #> <input timeout>o
<basic page #>:=	<basic page # opc> <string>
<input timeout>:=	<input timeout opc> <INTEGER> { in seconds }
<cap request>:=	<data storage opc>o <local types opc>o <user language opc>o <system information opc>o
<cap response>:=	<version> <term cap list>
<term cap list>:=	<local storage info>o <data type info>o <user language>o <system information>o
<local storage info>:=	<local storage opc> <true> <false>
<data type info>:=	<data type opc> <text type> <still picture> <audio>o <graphic>o <moving picture>o <audio visual>o
<user language>:=	<user language opc>o <string> { list of alphabetical language codes encoded according to Annex B/ISO 639 (example: Dutch language: "NL") }
<system information>:=	<system information opc>o <string> <text type>:= <text type opc> <text type list>+ <end of list>
<still picture>:=	<still picture opc> <still picture list>+ <end of list>
<graphic>:=	not yet defined.
<audio>:=	<audio opc> <audio type list>+ <end of list>
<moving picture>:=	<moving picture opc> <moving picture list>+ <end of list>
<audio visual>:=	<audio visual opc> <audio visual list>+ <end of list>
<text type list>:=	<INTEGER : 0> { T.51String } <INTEGER : 1> { VEMMI high quality text } <INTEGER : 2> { ISO 8859-1 } <INTEGER : 3> { ISO 10646-1 } <INTEGER : 4> { Shift JIS code } <INTEGER : 5> { ISO 8859-2 } <INTEGER : 6> { ISO 8859-3 } <INTEGER : 7> { ISO 8859-4 } <INTEGER : 8> { ISO 8859-5 } <INTEGER : 9> { ISO 8859-6 } <INTEGER : 10> { ISO 8859-7 } <INTEGER : 11> { ISO 8859-8 } <INTEGER : 12> { ISO 8859-9 } <INTEGER : 13> { ISO 8859-10 }

<error type>:= <INTEGER : 0> | { *general error* }
 <INTEGER : 1> | { *unknown command* }
 <INTEGER : 2> | { *erroneous command* }
 <INTEGER : 3> | { *object syntax error* }
 <INTEGER : 4> | { *unexpected command* }
 <INTEGER : 5> | { *out of memory* }
 <INTEGER : 6> | { *cannot process audio objects* }
 <INTEGER : 7> | { *cannot process video objects* }
 <INTEGER : 8> | { *invalid colour index* }
 <INTEGER : 9> | { *file not found* }
 <INTEGER : 10> | { *conversion to bitmap failed* }
 <INTEGER : 11> | { *cannot process direct col. definition* }
 <INTEGER : 12> | { *operative object has been rejected by the user* }
 <INTEGER : 13> | { *out of local storage space* }
 <INTEGER : 14> | { *a closed object has been destroyed* }
 <INTEGER : 15> | { *service not supported* }

<error oin>:= <error oin opc> <INTEGER>

<error cin>:= <error cin opc> <INTEGER>

<error com code>:= <error com code opc> <a/b> { *a = 2..4, b = 0..15* }

<transfer entity>:= <data transfer> | <transfer abort>

<data transfer>:= <INTEGER : 0> <transfer id> <current block number> <current block data length>
 <block>

<transfer id>:= <INTEGER> { *unique transfer identifier for each file* }

<current block number>:= <INTEGER> { *from one to the total number of blocks* }

<current block data length>:= <INTEGER>

<block>:= <file&block> | <current block>

<file&block>:= <filespec> <current block>

<filespec>:= <filename> <creation date> <file length> <total number of blocks>

<creation date>:= <string> { *in the format YYYYMMDDHHMMSS* }

<file length>:= <INTEGER>

<total number of blocks>:= <INTEGER>

<transfer abort>:= <INTEGER : 1> <transfer id>

<current block>:= <current block opc> <current block data>

<current block data>:= <a/b>+ { *a, b = 0..15* }

8.5 Objetos y componentes

<display object>:= <application bar> | <button bar> | <pop-up menu> |
 <dialogue box> | <operative object> | <sound object> |
 <video object> | <message box>

<application bar>:=	<appl bar opc> <application bar body> <menu item>+
<application bar body>:=	<xpos>o <ypos>o <height>o <width>o <closed>o <notaccessible>o <defactive>o <vertical>o
<menu item>:=	<bar menu choice> <pull down menu item>*
<bar menu choice>:=	<bar menu choice opc> <menu choice>
<pull down menu item>:=	<pull down menu choice> <cas menu choice>
<pull down menu choice>:=	<pull down menu choice opc> <menu choice> <separated>o
<cas menu choice>:=	<cas menu choice opc> <menu choice> <separated>o
<menu choice>:=	<cin> <height>o <width>o <notaccessible>o <text> <locactact>o <locactval>o
<button bar>:=	<button bar opc> <button bar body> <push button comp>+
<button bar body>:=	<xpos> <ypos> <width>o <height>o <closed>o <notaccessible>o <vertical>o <defactive>o <modal>o
<push button comp>:=	<push button comp opc> <cin> <height>o <width>o <closed>o <notaccessible>o <button> <locactact>o <locactval>o
<pop-up menu>:=	<pop-up menu opc> <pop-up menu body> <pop-up menu choice>+
<pop-up menu body>:=	<xpos> <ypos> <width>o <closed>o <notaccessible>o <title>o <title font>o <defactive>o <modal>o
<pop-up menu choice>:=	<pop-up menu choice opc> <menu choice> <separated>o <cas menu choice>*
<dialogue box>:=	<dialogue box opc> <dialog box body> <separator>* <frame>* <graphic output area>* <text area>* <text input field>* <box push button>* <check box>* <radio button>* <list box>* <combo box>* <slider>* <sensitive area>* <text component>* <sensitive text>*
<dialog box body>:=	<xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <no border>o <title>o <defactive>o <modal>o <maximizable>o <background>o <store ini values>o
<background>:=	<colour> <bitmapped>
<bitmapped>:=	<bin reference> <bitmap disptype>o <colour>o
<separator>:=	<separator opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <vertical>o <colour>o
<frame>:=	<frame opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <colour>o
<graphic output area>:=	<graphic output area opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <graphic type>
<graphic type>:=	<bitmap> <videotex>
<bitmap>:=	<bitmap disp type>o <bin reference>
<videotex>:=	<data syntax type> <direct videotex content> <vin reference>
<data syntax type>:=	<INTEGER : 0> { Annex B/T.101 } <INTEGER : 1> { Annex C/T.101 } <INTEGER : 2> { Annex D/T.101 }
<direct videotex content>:=	<videotex content opc> <nr of vtx bytes> <vtx bytes>+
<nr of vtx bytes>:=	<INTEGER> { number of Videotex bytes }

<text area>:=	<text area opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <initial font>o <maxim text>o <no scrolling tools>o <no format>o <no border>o <autoscroll>o <text area content>o
<text area content>:=	<cin reference> { <i>cin of a text component</i> } <in-text>+
<text component>:=	<text component opc> <cin> <prev text>o <current text> <next text>o
<prev text>:=	<prev text opc> <cin reference> { <i>cin of a text component</i> }
<next text>:=	<next text opc> <cin reference> { <i>cin of a text component</i> }
<current text>:=	<current text opc> <tin reference> { <i>tin of text resource object</i> } <in-text>+
<in-text>:=	<text> <in-text string>
<in-text string>:=	<fin reference> { <i>fin of a font resource object</i> } <sensitive text reference>
<sensitive text reference>:=	<cin reference> <text> { <i>cin of a sensitive text component</i> }
<sensitive text>:=	<sensitive text opc> <cin> <notaccessible>o <locactact>o <locactval>o
<text input field>:=	<text input field opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <maxim text>o <default text>o <label>o <label font>o <input type>o <echo type>o <echo char>o <max char>o <max line>o <multiline>o <input transform>o <locactact>o <locactval>o
<box push button>:=	<box push button opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <button> <locactact>o <locactval>o
<check box>:=	<check box opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <label>o <label font>o <marked>o <locactact>o <locactval>o
<radio button>:=	<radio button opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <label>o <label font>o <group>o <marked>o <locactact>o <locactval>o
<list box>:=	<list box opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <default item>o <sorted>o <multiple choice>o <locactact>o <locactval>o <list text unit>*
<combo box>:=	<combo box opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <default item>o <sorted>o <no drop>o <no consistency>o <no edit>o <locactact>o <locactval>o <list text unit>*
<list text unit>:=	<list index>o <text> <icon reference>o <item locact>o <item locval>o
<icon reference>:=	<bin reference>
<slider>:=	<slider opc> <cin> <xpos>o <ypos>o <height>o <width>o <closed>o <notaccessible>o <label>o <label font>o <negative>o <min value> <negative>o <max value> <increment>o <initial info>o <direct in>o <vertical>o <locactact>o <locactval>o
<initial info>:=	<negative>o <initial value>
<sensitive area>:=	<sensitive area opc> <cin> <xpos>o <ypos>o <width>o <height>o <closed>o <notaccessible>o <locator>o <locactact>o <locactval>o
<operative object>:=	<operative object opc> <closed>o <program name> <program filename> <program description> <program about> <program parameter>o <program type>
<sound object>:=	<sound object opc> <closed>o <filename> <sound format>
<sound format>:=	<INTEGER : 0> { <i>wave format</i> } <INTEGER : 1> { <i>MIDI format</i> }

<video object>:= <video object opc> <closed>o <filename> <moving picture list>o
 <message box>:= <message box opc> <xpos>o <ypos>o <width>o <height>o <closed>o <no border>o <title>o <modal>o <message type>o <max time>o <lifetime>o <no sound>o <attributed text>+
 <message type>:= <message type opc> <message>
 <message>:= <INTEGER : 0> | { *general message* }
 <INTEGER : 1> | { *information message* }
 <INTEGER : 2> | { *warning message* }
 <INTEGER : 3> | { *action message* }
 <lifetime>:= <lifetime opc> <destroy event>
 <destroy event>:= <INTEGER : 0> | { *destroy by any user action* }
 <INTEGER : 1> | { *destroy by validation of implicit defined button* }
 <INTEGER : 2> | { *destroy by a command from the host* }
 <INTEGER : 3> | { *close by validation of implicit defined button* }
 <INTEGER : 4> | { *close by any user interaction* }
 <INTEGER : 5> | { *no implicit lifetime defined* }
 <max time>:= <max time opc> <INTEGER> { *time period in seconds* }

<resource object>:= <bitmap object> <text object> <videotex object>
--

<bitmap object>:= <bitmap object opc> <bitmap data> | <bitmap file>
 <bitmap data>:= <INTEGER : 0> <bmwidth> <bmheight> <bmcompr>o <pixel definition>
 <bmwidth>:= <INTEGER>
 <bmheight>:= <INTEGER>
 <bmcompr>:= <bmcompr opc> <INTEGER : 0> { *no compression* }
 <pixel definition>:= <index def> | <rgb def>
 <index def>:= <index def opc> <bits per pixel>o <colour entry>o <colour list>
 <bits per pixel>:= <bits per unit>
 <colour entry>:= <colour entry opc> <INTEGER>
 <colour list>:= <list spec> { *bmwidth * bmheight integers* }
 <rgb def>:= <rgb def opc> <bits per component>o <component list>
 <bits per component>:= <bits per unit>
 <component list>:= <list spec> { *3 * bmwidth * bmheight integers* }
 <bits per unit>:= <bits per unit opc> <INTEGER>
 <bitmap file>:= <INTEGER : 1> <filename> <pict file type>
 <pict file type>:= <INTEGER : 0> | { *JPEG* }
 <INTEGER : 1> | { *GIF* }
 <INTEGER : 2> | { *BMP* }
 := <fn family>o <fn height>o <fn bold>o <fn underline>o <fn italic>o <fn colour>o
 <fn family>:= <fn family opc> <family>

<family>:= <INTEGER : 0> | { *SWISS* }
 <INTEGER : 1> | { *ROMAN* }
 <INTEGER : 2> { *FIXFONT* }

 <fn height>:= <fn height opc> <INTEGER>
 <fn bold>:= <fn bold opc>
 <fn underline>:= <fn underline opc>
 <fn italic>:= <fn italic opc>
 <fn colour>:= <fn colour opc> <INTEGER>

 <text object>:= <text object opc> <in-text>+ | <text file>
 <text file>:= <file name> <text file type>
 <text file type>:= <INTEGER : 0> | { *"in-text" format* }
 <INTEGER : 1> { *plain text, incl. CR, LF* }

 <videotex object>:= <videotex object opc> <vtx file>
 <videotex file>:= <filename> <data syntax type>

<metacode object>:= <metacode object opc> <command>+ <command end>

8.6 Acciones locales

<locactact>:= <locactact opc> <local action>+
 <locactval>:= <locactval opc> <local action>+
 <local action>:= <report command> | <general command> | <specific command>
 <report command>:= <loc command opc> <report type>
 <report type>:= <INTEGER : 0> | { *report OIN, CIN* }
 <INTEGER : 1> | { *report current value* }
 <INTEGER : 2> | { *report all values* }
 <INTEGER : 3> { *report all changed values* }

 <general command>:= <loc command opc> <general command type>
 <general command type>:= <INTEGER : 50> | { *user lock* }
 <INTEGER : 51> { *set initial values and states in the parent object* }

 <specific command>:= <loc command opc> <specific command type> <oin spec> | <cin spec>
 <specific command type>:= <INTEGER : 100> | { *open components of the parent object* }
 <INTEGER : 101> | { *close components of the parent object* }
 <INTEGER : 102> | { *open objects* }
 <INTEGER : 103> | { *close objects* }
 <INTEGER : 104> | { *change components of the parent object to inaccessible* }
 <INTEGER : 105> | { *change components of the parent object to accessible* }
 <INTEGER : 106> | { *destroy objects* }
 <INTEGER : 107> { *open blocking object (only one)* }

9 Codificación

9.1 Estructura de las instrucciones

La PDE de una entidad de imagen lleva una o más instrucciones VEMMI. La última instrucción VEMMI de una PE se puede dividir en una o más PE. Esto se indica mediante el primer byte de PDE, llamado «indicador más datos». La sintaxis de PDE es:

<PDE>:= <mdi> <command>+

<mdi>:= <2/14> | { *the last command is only partly contained in this PE, it continues in the next PE; "MORE"* }
 <2/15> { *the last command is completely contained in the PE; "LAST"* }

NOTA – La separación de instrucciones en más de una PE no aparece en la especificación de la gramática.

Varias instrucciones VEMMI comienzan por un encabezamiento VEMMI. Ese encabezamiento se utiliza para identificar los datos VEMMI en un tren de datos de la aplicación estándar. A esa secuencia de datos VEMMI se la denomina Entidad de datos. La última instrucción de una entidad de datos puede dividirse entre una o más entidades de datos. Esto lo indica el MDI (indicador de más datos). Ejemplos:

Tres instrucciones contenidas en una entidad de datos:

Encabezamiento VEMMI	MDI (último)	Entidad de datos		
		Instrucción 1	Instrucción 2	Instrucción 3

Tres instrucciones en dos entidades de datos, una dividida:

Encabezamiento VEMMI	MDI (más)	Entidad de datos		Encabezamiento VEMMI	MDI (último)	Entidad de datos	
		Instrucción 1	Instrucción 2 (primera parte)			Instrucción 2 (última parte)	Instrucción 3

9.2 Estructura de objeto, componente y atributo

Estructura de objeto

Los objetos tienen una de las estructuras siguientes:

- a) <object code> <object body> <component>*
- b) <object code> <attribute>+
- c) <object code> <command>+

Estructura de componente

<component code> <cin> <attribute>*

Estructura de atributo

Los atributos tienen una de las estructuras siguientes:

- a) <attribute code> { el atributo es definido sólo por el código };
- b) <attribute code, value> { el atributo es definido por el código y el valor };
- c) <attribute> { el atributo es definido por otro atributo };
- d) <attribute code, attribute> { el atributo es definido por el código y por otro atributo }.

9.3 Codificación de símbolos de terminal

9.3.1 Códigos operacionales

Los códigos operacionales (opcodes) se utilizan para identificar las partes de las instrucciones VEMMI. Se emplean para codificar:

- códigos de instrucción;
- códigos de objeto;
- códigos de componente;
- el código MDI;
- códigos de atributo.

La notación de símbolo de terminal es: <a/b> con a, b = 0 .. 15 (notación hexadecimal).

9.3.2 Enteros

Este formato se utiliza para codificar valores no negativos sin fracciones. Un entero consiste en una secuencia de byte de uno o más bytes.

Reglas de codificación:

- a) bit 8 puesto a 1 significa no extensión, si es el último byte de la secuencia;
- b) bit 8 puesto a 0 y bit 7 puesto a 1 indican una extensión, si no es el último bit;
- c) el valor codificado se especifica:
 - 1) para enteros de un byte, por los bits 7 a 1;
 - 2) para enteros de varios bytes, por los bits 6 a 1 de todos los bytes, salvo el último, y por los bits 7 a 1 del último byte;
- d) la parte más significativa del valor se codifica en el primer byte;
- e) la parte menos significativa del valor se codifica en el último byte.

Símbolos que emplean enteros:

<INTEGER>

<false>:= <INTEGER : 0>

<true>:= <INTEGER : 1>

Ejemplo: Representación codificada del número 37.

b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
1	0	1	0	0	1	0	1
No extensión	Bits de valor entero						

Ejemplo: Representación codificada del número 1741.

Byte 1	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
	0	1	0	0	1	1	0	1
	Extensión			Bits más significativos				

Byte 2	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
	1	1	0	0	1	1	0	1
	No extensión	Bits menos significativos						

9.3.3 Enumerado

Este tipo de datos designa un valor de un conjunto de valores normalizados.

Se codifican como enteros con la notación: <INTEGER : n> donde n es un valor decimal.

9.3.4 Cadenas

Este tipo de datos designa una secuencia de caracteres con la siguiente estructura:

<string>:=	<latin string> <non latin string> <unicode string>
<latin string>:=	<character>+ <end of string>
<character>:=	<a/b> <c/d> <CR> <LF> <ESC> { a = 2..7, b = 0..15, c = 10, 15 characters coded according to ISO 8859 or T.51String }
<end of string>:=	<9/12> { ST (string terminator) from ISO 6429 }
<CR>:=	<0/13>
<LF>:=	<0/10>
<non latin string>:=	<shift JIS string> <T.52 string>
<shift JIS string>:=	<a/b> <c/d> <e/f> <CR> <LF> { a = 2..7, 10..13; b = 0..15, c = 8,9,14,15; d = 0..15; e = 4..15; f = 0..15 }
<end of JIS string>:=	<1/11> <5/12> { ST (string terminator) from ISO 6429 }
<T.52 string>:=	for further study
<unicode string>:=	<uni char> <uni char> <end of unicode string> { string according to ISO 10646-1 (BMP) }
<uni char>:=	<a/b> { a = 0..15, b = 0..15 }
<end of unicode>:=	<ST version> <FF version>
<ST version>:=	<0/0> <9/12>
<FF version>:=	<F/F> <F/F>

9.3.5 NDC

Este tipo de dato designa valores escalares horizontales o verticales en el espacio NDC. Un NDC es un número real de punto fijo con la gama 0.0 (incluido) a 1.0 (excluido).

Reglas de codificación:

- a) un valor NDC se codifica en una secuencia de uno o más bytes;
- b) bit 8 puesto a 1 significa no extensión, si es el último byte de la secuencia;
- c) bit 8 puesto a 0 y bit 7 puesto a 1 indican una extensión, si no es el último byte;
- d) el valor codificado se especifica:
 - 1) para un byte NDC, por los bits 7 a 1;
 - 2) para varios bytes NDC, por los bits 6 a 1 de todos los bytes excepto el último, y por los bits 7 a 1 del último byte;
- e) el punto radix binario no se codifica, pero se supone que es:
 - 1) para un byte NDC, el próximo a la izquierda del bit 7;
 - 2) para varios bytes NDC, el próximo a la izquierda del bit 6 del primer byte;
- f) la parte más significativa del valor se codifica en el primer byte;
- g) la parte menos significativa del valor se codifica en el último byte.

La notación de símbolo de terminal es: <NDC>.

Ejemplo: Representación codificada del número $0,3125_{10} = 0,0101_2$

b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
1	0	1	0	1	0	0	0
No extensión	Bits de valor NDC						

Ejemplo: Representación codificada del NDC $0,205078125_{10} = 0,001101001_2$

Byte 1	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
	0	1	0	0	1	1	0	1
	Extensión		Bits más significativos					

Byte 2	b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁
	1	0	0	1	0	0	0	0
	No extensión	Bits menos significativos						

9.4 Atributos y símbolos de nivel inferior

<autoscroll>	:=	<6/9>
<attributed text>	:=	<text> <fin reference>
<autostore>	:=	<7/8>
<bin>	:=	<INTEGER>
<bin reference>	:=	<11/8> <bin>
<bitmap disp type>	:=	<11/11> <INTEGER : 0> { <i>stretched</i> } <INTEGER : 1> { <i>centred</i> } <INTEGER : 2> { <i>tiled</i> }
<block index>	:=	<12/4> <INTEGER>
<button>	:=	<text> <bin reference>
<cin>	:=	<INTEGER>
<cin reference>	:=	<12/7> <INTEGER> { <i>reference to a component</i> }
<cin spec>	:=	<in spec>+
<closed>	:=	<6/8>
<colour>	:=	<11/10> <INTEGER> { <i>colour index</i> }
<command end>	:=	<2/15>
<defactive>	:=	<11/7> <INTEGER> { <i>the referenced cin</i> }
<default item>	:=	<12/3> <INTEGER>
<default text>	:=	<text>
<direct in>	:=	<7/10>
<display character>	:=	<a/b> { <i>a = 2..15, b = 0..15</i> }
<echo char>	:=	<14/6> <display character>
<echo type>	:=	<12/0> <INTEGER : 0> { <i>local echo</i> } <INTEGER : 1> { <i>no echo</i> } <INTEGER : 2> { <i>echo defined character</i> }
<end of list>	:=	<3/15>
<filename>	:=	<14/8> <string>
<fin>	:=	<INTEGER>
<fin reference>	:=	<10/13> <INTEGER> { <i>reference to a font resource object</i> }
<group>	:=	<12/2> <INTEGER>
<height>	:=	<9/2> <NDC>
<in spec>	:=	<list spec> <range spec> <one in> { <i>defines a list, a range or a single identification number</i> }
<list spec>	:=	<11/0> <INTEGER>+ <end of list>
<range spec>	:=	<13/0> <from> <to>
<from>	:=	<INTEGER>
<to>	:=	<INTEGER>
<one in>	:=	<13/5> <INTEGER>
<increment>	:=	<12/10> <INTEGER>
<initial font>	:=	<11/12> <INTEGER> { <i>the referenced fin</i> }
<initial value>	:=	<12/11> <INTEGER>
<input type>	:=	<11/14> <INTEGER : 0> { <i>any character</i> } <INTEGER : 1> { <i>alphabetic</i> } <INTEGER : 2> { <i>numeric</i> } <INTEGER : 3> { <i>alphanumeric</i> }
<input transform>	:=	<11/5> <INTEGER : 0> { <i>no transformation</i> } <INTEGER : 1> { <i>to lower</i> } <INTEGER : 2> { <i>to upper</i> }
<item locact>	:=	<7/13> <local action>+
<item locval>	:=	<7/14> <local action>+
<label font>	:=	<11/13> <INTEGER> { <i>the referenced fin</i> }
<label>	:=	<14/5> <string>
<list index>	:=	<12/4> <INTEGER>
<locator>	:=	<6/0>
<marked>	:=	<7/1>
<max char>	:=	<12/1> <INTEGER>

<max line>	:=	<12/5>	<INTEGER>
<max value>	:=	<12/9>	<INTEGER>
<maxim text>	:=	<7/0>	
<maximizable>	:=	<6/14>	
<min value>	:=	<12/8>	<INTEGER>
<modal>	:=	<6/12>	
<multiline>	:=	<6/15>	
<multiple choice>	:=	<7/2>	
<negative>	:=	<7/9>	
<no border>	:=	<6/13>	
<no consistency>	:=	<7/15>	
<no drop>	:=	<7/3>	
<no edit>	:=	<7/4>	
<no format>	:=	<7/6>	
<no scrolling tools>	:=	<7/5>	
<no sound>	:=	<7/11>	
<notaccessible>	:=	<6/11>	
<num blocks>	:=	<10/8>	<INTEGER>
<oin>	:=	<INTEGER>	
<oin spec>	:=	<in spec>+	{ <i>defines object identifiers</i> }
<output height>	:=	<9/4>	<NDC>
<program about>	:=	<string>	
<program description>	:=	<string>	
<program filename>	:=	<filename>	
<program name>	:=	<string>	
<program parameter>	:=	<program parameter opc>	<string>
<program type>	:=	<13/1>	<INTEGER : 0> { <i>standalone program</i> }
			<INTEGER : 1> { <i>program with filter interface</i> }
<result>	:=	<true> <false>	
<scrolling tools>	:=	<11/1>	<INTEGER : 0> { <i>terminal dependent</i> }
			<INTEGER : 1> { <i>two column scrolling</i> }
			<INTEGER : 2> { <i>two row scrolling</i> }
<slider value>	:=	<INTEGER>	
<store result>	:=	<true> <false>	
<separated>	:=	<6/10>	
<sorted>	:=	<7/12>	
<template>	:=	<7/7>	
<text>	:=	<14/3>	<string>
<timestamp>	:=	<12/14>	<INTEGER>
<tin reference>	:=	<10/15>	<INTEGER> { <i>reference to a text resource object</i> }
<title font>	:=	<11/9>	<INTEGER> { <i>the referenced fin</i> }
<title>	:=	<14/4>	<string>
<vertical>	:=	<6/7>	
<vin reference>	:=	<10/12>	<INTEGER>
<width>	:=	<9/3>	<NDC>
<xloc>	:=	<NDC>	
<xpos>	:=	<9/0>	<NDC>
<yes/no>	:=	<true> <false>	
<ypos>	:=	<9/1>	<NDC>
<yloc>	:=	<NDC>	
<vtx byte>	:=	<a/a>	{ <i>a = 0..15</i> }

9.5 Códigos operacionales (opcódigos)

En los Cuadros 39 y 40 se suprime el sufijo «opc» de los símbolos de la especificación de la sintaxis de las instrucciones.

Las posiciones de código que actualmente no se utilizan se reservan para futuras ampliaciones.

CUADRO 39/T.107

Opcódigos de las instrucciones de anfitrión

	2	3
0	open	create object
1	close	delete outdated objects
2	resume	modify component (véase también 9.6)
3	suspend	obj location change
4	close all	load coltable
5	reset col	open application
6	user lock	store objects
7	user unlock	erase objects
8	open object	open blocking object
9	close object	resource transfer
10	destroy object	term cap request
11	access disable	
12	access enable	set options
13	translation mode	
14	(véase la Nota)	
15	(véase la Nota)	

NOTA – Estas posiciones de código se reservan para el MDI.

CUADRO 40/T.107

Opcódigos de instrucciones de terminal

	2	3
0	store objects resp	
1	object retrans	
2	open appl resp	
3	user data	
4	error	
5		
6	term cap resp	
7		
8		
9		resource transfer
10		
11		
12		
13	translation mode	
14		
15		

NOTA – Los Cuadros 41 a 43 contienen símbolos que tienen el mismo código. Sus nombres se separan mediante «/».

CUADRO 41/T.107

Opcódigos (Parte 1)

	Objetos	Códigos «fn»	Componentes	
	2	3	4	5
0	application bar		bar menu choice	slider
1	button bar		pull-down choice	text component
2	pop-up menu		push button comp	sensitive text
3	dialogue box		pop-up choice	caso menu choice
4			separator	
5	operative object		frame	
6	sound object		graphic output area	
7	video object		text area	
8	message box		text input field	
9	metacode object		box push button	
10	bitmap object		check box	
11	font object		radio button	
12	text object		list box	
13	videotex object		combo box	
14		command end	sensitive area	
15		end of list		

CUADRO 42/T.107

Opcódigos (Parte 2)

	Atributos booleanos		Coordenadas, dimensiones
	6	7	9
0	fn bold/locator	maxim text	xpos
1	fn underline/local storage	marked	ypos
2	fn italic	multiple choice	height
3	index def	no drop	width
4	rgb def	no edit	output height
5	locactact	no scrolling tools	
6	locactval	no format	
7	vertical	template	
8	closed	autostore	
9	autoscroll	negative	
10	separated	direct in/more blocks	
11	notaccessible	no sound/store ini values	
12	modal	sorted	end of string
13	no border	item locact	
14	maximizable	item locval	
15	multiline	no consistency	

CUADRO 43/T.107

Opcódigos (Parte 3)

	Atributos de entero				Atributos de cadena
	10	11	12	13	14
0	fn family	list spec	echo type	range spec	applid/ user language
1	fn height/ data types	error oin/ scrolling tools	max char	program type	appl add data/ system information
2	fn colour/ text types	error cin	group	prev text	
3	bmcompr/ still picture	error com code	default item	current text	text
4	bits per unit	local commands	list index/ block index	next text	title
5	colour entry/ audio	input transform	max line	one in	label
6	moving picture				echo char
7	audiovisual	defactive	cin reference		vtx content
8	num blocks	bin reference	min value		filename
9	message type	title font	max value		program parameter
10	lifetime	colour	increment		current block
11	default vtx	bitmap disptype	initial value		
12	vin reference	initial font	max time		basic page #
13	fin reference	label font	input timeout		
14		input type	timestamp		
15	tin reference				

9.6 Sintaxis de VEMMI_Modificar_componente

VEMMI_Modificar_componente se utiliza para modificar o añadir uno o más objetos de un componente. Esta instrucción tiene la misma sintaxis que VEMMI_Crear_objeto, salvo que contiene sólo aquellos atributos que habrá que modificar. A continuación se enumeran todos los atributos que son modificables o sus definiciones de símbolo superior:

autoscroll	echo type	list text unit	no sound
attributed text	group	locactval	notaccessible
bin reference	increment	locactact	previous text
cin reference	initial font	max char	text
closed	initial info	max value	tin reference
colour	input type	marked	title
current text	intext	message	
default item	label	min value	
default text	label font	negative	
echo char	lifetime	next text	

La modificación de los atributos booleanos se consigue transmitiendo nuevamente el mismo atributo. La aplicación anfitriona puede fijar o no fijar el valor de un atributo booleano.

Para fijar un atributo booleano, la aplicación anfitriona transmitirá ese atributo una vez (esto equivale a la sintaxis de creación de objeto). Para desfijar un atributo booleano, la aplicación anfitriona transmitirá ese atributo dos veces, una inmediatamente después de la otra.

En el ejemplo siguiente, un menú superpuesto ofrece cuatro opciones al usuario, dos accesibles (CIN 35 y 36) y dos inaccesibles (CIN 37 y 38):

<u>C</u> omenzar nuevo juego
<u>N</u> ivel principiante
<u>N</u> ivel intermedio
<u>N</u> ivel avanzado

Al recibir la instrucción VEMMI_Modificar_componente con:

```
<oin : 2>  
<cin : 37> <NotAccessible> <NotAccessible>  
<cin : 39> <text : "&Double Cube"> <separated>
```

cambia el menú: un componente inaccesible se convierte en accesible, y se añade un componente nuevo.

<u>C</u> omenzar nuevo juego
<u>N</u> ivel principiante
<u>N</u> ivel intermedio
<u>N</u> ivel avanzado
<u>D</u> oble cubo

Cuando el atributo es opcional, se aplican los valores por defecto en el caso de que VEMMI_Modificar_componente añada un componente nuevo. Esta es la misma regla que utiliza VEMMI_Crear_objeto. Si VEMMI_Modificar_componente referencia un componente que ya existe, sólo se aplicarán los atributos transmitidos. Los atributos faltantes no cambian.

No se aplicará VEMMI_Modificar_componente a los componentes de la barra de aplicaciones.

En 7.6.1.11 y 7.6.1.12 se describen los requisitos específicos para modificar los ítems de lista.

9.7 Valores por defecto

Los valores por defecto se aplican cuando un atributo opcional no está contenido en la definición de elemento. Como excepción, VEMMI_Modificar_componente se puede aplicar a componentes existentes, véase 9.6. Los valores por defecto dependen del atributo de tipo de dato y del objeto o componente al que pertenecen.

Para el valor por defecto de un atributo booleano (estos se codifican únicamente con opcódigos), se supondrá el significado negativo lógico comparado con su definición. Un valor por defecto de un atributo booleano no es codificable, y no aparecerá nunca en la definición de elemento. Véanse los Cuadros 44 a 48.

CUADRO 44/T.107

Valores por defecto de atributos booleanos

Atributo	Valor por defecto	Atributo	Valor por defecto
fn bold	normal, non-bold	maxim text	no maximised text
fn underline	not underlined	marked	unmarked
fn italic	non-italic	multiple choice	single choice
vertical	horizontal	no drop	drop
closed	open	no edit	editable
notaccessible	accessible	no scrolling tools	with scrolling tools
modal	non-modal	no format	formatable
no border	with border		
autoscroll	no autoscroll	template	no "template" store
separated	not separated	autostore	no "autostore"
multiline	single line	no sound	with sound
no consistency	with consistency	sorted	not sorted
maximizable	not maximizable	negative	positive
		direct in	no direct input

CUADRO 45/T.107

Valores por defecto de atributos enteros

Atributo	Valor por defecto	Atributo	Valor por defecto
fn family	SWISS	input type	0
fn height	10	echo type	0
fn colour	0 (black)	max char	1
bits per units (pixel)	1	group	1
bits per units (component)	8	default item	1
		list index	1 for the first, all other are numbered consecutively
message type	0	max line	terminal dependent
lifetime	1	sound format	0
colour entry	0	initial value	min value
		increment	1
defactive	1	max time	10
title font	default font	input timeout	infinite
colour	0 (black)		
bitmap disptype	stretched		
initial font	default font		
label font	default font		

CUADRO 46/T.107

Valores por defecto de atributos NDC

Atributo	Valor por defecto
xpos, ypos of the message box	terminal dependent
xpos	0,0
ypos	0,0

CUADRO 47/T.107

Valores por defecto de anchura y altura

Elemento	Valor por defecto de anchura	Valor por defecto de altura
barra de aplicaciones	dependiente del terminal	dependiente del terminal
botón; barra de botones	0,125	0,03
menú superpuesto	dependiente del terminal	
caja de diálogo	1,0	0,75
separador	anchura de casilla de diálogo	altura de casilla de diálogo
marco	anchura de casilla de diálogo	altura de casilla de diálogo
zona de texto	anchura de casilla de diálogo	altura de casilla de diálogo
zona de mapa de bits	anchura de casilla de diálogo	altura de casilla de diálogo
campo de entrada de texto	anchura de casilla de diálogo	0,03
botón pulsado	0,125	0,03
casilla de comprobación	0,125	0,03
botón radio	0,125	0,03
caja de lista	anchura de casilla de diálogo	altura de casilla de diálogo
caja de combinación	anchura de casilla de diálogo	altura de casilla de diálogo
zona sensible	0,125	0,03
caja de mensaje	dependiente del terminal	dependiente del terminal

CUADRO 48/T.107

Valores por defecto de atributos de cadena

Atributo	Valor por defecto
echo char	2/13

10 Introducción del servicio VEMMI en las Recomendaciones existentes relativas a videotex

10.1 Introducción de VEMMI en la Recomendación T.101 [4]

Los elementos de protocolo VEMMI se hacen corresponder con los elementos de datos de presentación videotex (VPDEs).

Para habilitar el flujo de datos entre el terminal videotex y la aplicación VEMMI, el canal lógico será transparente para los bloques de datos VEMMI.

10.2 Introducción de VEMMI en la Recomendación T.105 [6]

Los elementos de protocolo VEMMI se hacen corresponder con uno o más elementos del servicio SBV_VTX_Datos.

Para habilitar el flujo de datos entre el terminal videotex y la aplicación VEMMI, el canal lógico será transparente para los bloques de datos VEMMI.

Anexo A

Cadena T.51

(Este anexo es parte integrante de la presente Recomendación)

A.1 Alcance

Este anexo describe las reglas que se observarán al codificar nombres, cadenas, etc., mediante la Recomendación T.51 [2]. La cadena T.51 que define este anexo sirve como modelo de referencia para restringir esta Recomendación a los elementos de los mecanismos de extensión de código de los juegos de caracteres y el repertorio que se necesitan para facilitar las realizaciones.

Esta definición no es específica de un servicio telemático particular, pero puede ser referenciada por las normas de aplicaciones telemáticas.

A.2 Juegos de caracteres gráficos

El principal juego de caracteres gráficos (véase la Figura A.1) es idéntico al juego de caracteres gráficos de la Versión Internacional de Referencia (IRV) del juego de caracteres codificado con 7 bits de la Recomendación T.50 [1].

				b ₇	0	0	0	0	1	1	1	1	
				b ₆	0	0	1	1	0	0	1	1	
				b ₅	0	1	0	1	0	1	0	1	
					0	1	2	3	4	5	6	7	
b ₄	b ₃	b ₂	b ₁										
0	0	0	0	0				0	@	P	.	p	
0	0	0	1	1				!	1	A	Q	a	q
0	0	1	0	2				"	2	B	R	b	r
0	0	1	1	3				#	3	C	S	c	s
0	1	0	0	4				\$	4	D	T	d	t
0	1	0	1	5				%	5	E	U	e	u
0	1	1	0	6				&	6	F	V	f	v
0	1	1	1	7				'	7	G	W	g	w
1	0	0	0	8				(8	H	X	h	x
1	0	0	1	9)	9	I	Y	i	y
1	0	1	0	10				*	:	J	Z	j	z
1	0	1	1	11				+	;	K	[k	{
1	1	0	0	12				,	<	L	\	l	
1	1	0	1	13				-	=	M]	m	}
1	1	1	0	14				.	>	N	^	n	~
1	1	1	1	15				/	?	O	_	o	

T0821490-95/d42

FIGURA A.1/T.107
Principal juego de caracteres gráficos para una cadena T.51

El juego de caracteres gráficos suplementario se especifica en la Figura A.2.

					b ₇	0	0	0	0	1	1	1	1
					b ₆	0	0	1	1	0	0	1	1
					b ₅	0	1	0	1	0	1	0	1
						0	1	2	3	4	5	6	7
b ₄	b ₃	b ₂	b ₁										
0	0	0	0	0			NBSP	◦		—	Ω	Κ	
0	0	0	1	1			ı	±	˘	ı	Æ	æ	
0	0	1	0	2			¢	²	˙	®	Ð	ð	
0	0	1	1	3			£	³	ˆ	©	ª	ä	
0	1	0	0	4				x	˜	™	℥	℥	
0	1	0	1	5			¥	μ	—	♯		ı	
0	1	1	0	6				¶	˘	¬	ıı	ıı	
0	1	1	1	7			§	•	•	ı	Ł	ł	
1	0	0	0	8			¤	÷	˙˙		Ł	ł	
1	0	0	1	9			˙	˙			Ø	ø	
1	0	1	0	10			“	”	°		Œ	œ	
1	0	1	1	11			«	»	˘		º	ß	
1	1	0	0	12			←	¼		⅛	ƒ	ƒ	
1	1	0	1	13			↑	½	”	⅜	ƒ	t	
1	1	1	0	14			→	¾	˘	⅝	ŋ	ŋ	
1	1	1	1	15			↓	ı	˘	⅞	˘n	SHY	

T0821500-95/d43


 Estas posiciones de código no se utilizarán

FIGURA A.2/T.107

Juego de caracteres gráficos suplementario para un cadena T.51

En una cadena T.51 se aplica lo siguiente:

- 1) El juego principal se designa como juego G0 y se invoca en la columna 2 a 7 del cuadro de códigos.
- 2) El juego suplementario se designa como juego G2. En el entorno de 8 bits, el juego se invoca en la columna 10 a 15.
- 3) En la cadena T.51 no se permiten secuencias de designación e invocación.
- 4) Todos los caracteres de la columna 4 del juego suplementario son caracteres sin espacio (diacríticos).
- 5) Las posiciones de código no asignadas están reservadas y no se utilizarán.
- 6) Las disposiciones de compatibilidad descritas en 2.2.4/T.51 [2] Notas 3, 4 y 6 no son aplicables.

A.3 Técnica de extensión de código

En el entorno de 8 bits no se permite la secuencia de extensión de código. En el entorno de 7 bits, se utiliza la función de un solo cambio SS2 (código 1/9) para invocar un carácter del juego G2. No se permiten las demás funciones de cambio.

A.4 Repertorio del juego de caracteres latinos

El repertorio de la cadena T.51 es idéntico al del superjuego del repertorio de juego de caracteres latinos especificado en el Anexo A/T.51 [2]. Se soportan todas las combinaciones de letras básicas con diacríticos especificadas en la Figura A.2/T.51 [2].

Las Notas y comentarios de A.4/T.51 [2] no son aplicables.

A.5 Funciones de control

No se autorizan las secuencias de invocación y designación relativas a funciones de control. Sólo se podrán utilizar los caracteres de control Retroceso del carro (CR, 0/13) y cambio de renglón (LF, 0/10) de las columnas 0 y 1 del cuadro de códigos vigente, especificadas en la Recomendación T.50 [1].

Anexo B

Subconjunto obligatorio de ISO 8859 [13]

(Este anexo no es parte integrante de la presente Recomendación)

	2	3	4	5	6	7		10	11	12	13	14	15
0	SP	0	@	P	'	p							
1	!	1	A	Q	a	q							
2	"	2	B	R	b	r							
3	#	3	C	S	c	s							
4	\$	4	D	T	d	t							
5	%	5	E	U	e	u							
6	&	6	F	V	f	v							
7	'	7	G	W	g	w							
8	(8	H	X	h	x							
9)	9	I	Y	i	y							
10	*	:	J	Z	j	z							
11	+	;	K	[k	{							
12	,	<	L	\	l								
13	-	=	M]	m	}							
14	.	>	N	^	n	~							
15	/	?	O	_	o								

FIGURA B.1/T.107

Subconjunto obligatorio del cuadro de códigos ISO 8859 [13]

Anexo C

Núcleo mínimo de tipos de datos

(Este anexo es parte integrante de la presente Recomendación)

Un terminal VEMMI soportará el siguiente núcleo mínimo de tipos de datos.

CUADRO C.1/T.107

Tipos de dato soportados

Clase	Norma de codificación	Observaciones
Para datos de texto		
	Texto de alta calidad VEMMI	
	ISO 8859-1 [13] (caracteres acentuados formados completamente) desde la posición 2/0 a la posición 7/15, como se indica en el Anexo B	Para caracteres latinos
	Código Shift JIS (para caracteres japoneses)	Para caracteres japoneses
Para imágenes fijas		
	ISO 10918 [16] (JPEG)	
	Mapa de bits independiente del dispositivo VEMMI	
	Formato de intercambio de gráficos (sm) Versión 98a. Compuserve Incorporated Columbus, Ohio, USA (sólo deben soportarse las funciones de la versión 87)	
	Mapa de bits independiente del dispositivo (DIB) Microsoft Windows con RLE4, RLE8 y 1, 4, 8 ó 24 bits por pixel	
Para datos de audio		
	Formato WAVE	(véase la Nota)
	Formato MIDI	(véase la Nota)
NOTA – Obligatorio en este caso significa que un terminal soportará todos los servicios VEMMI relacionados con la funcionalidad. No significa que un terminal estará equipado con el dispositivo necesario para visualizar estos datos.		