Capítulo 14

La documentación lingüística y la codificación de textos

Jost Gippert

Introducción

En la documentación lingüística, la notación de textos en forma escrita siempre ha desempeñado un papel importante, incluso después del desarrollo de los dispositivos de almacenamiento audiovisual. La era digital ha provocado, si acaso, un cambio menor a esta situación en tanto que ahora podemos esperar que nuestros datos escritos sean útiles para mucha gente y durante muchos siglos sin que necesariamente se impriman o se distribuyan como libros. Sin embargo, para lograr este propósito se debe tener en consideración ciertas cuestiones preliminares que se tratarán en este capítulo.

La representación de textos en forma digital difiere del uso de lápiz y papel pues presupone la adaptación de *códigos* claramente definidos en un sentido doble: la codificación de caracteres, es decir, de las letras en las palabras que habrán de escribirse, y la codificación de los elementos de la estructura textual, es decir, encabezados, ejemplos, listas de vocabulario, etc. Los dos tipos de codificación son cruciales para el intercambio de datos con otras personas: un futuro usuario que no tenga información sobre los esquemas de codificación que se hayan aplicado, probablemente tendrá grandes dificultades al tratar de volver a decodificar (y leer) lo que se escribió; en el peor de los casos, esos datos serán totalmente irrecuperables. En las páginas siguientes explicaré brevemente por qué esto es de esperarse y qué se puede hacer para evitarlo. Empezaremos con la codificación de las unidades de texto más pequeñas, es decir, los caracteres, y procederemos a elementos más grandes como palabras, frases y sintagmas. Otros tipos de codificación que pudieran caber en la presente discusión (especialmente la codificación de archivos; véase el capítulo 4) se tratarán de paso.

1. La codificación de caracteres: de 7 bits a 32 bits

1.1. Computadoras de Unidad Central (*Mainframe*): la era ASCII

En todos los equipos digitales modernos, la codificación de caracteres se basa en un conjunto determinado de correspondencias entre caracteres y valores numéricos, en el que cada caracter está representado por un valor único. Para codificar las 26 letras del alfabeto latino dos veces (mayúsculas y minúsculas), más los dígitos del 0 al 9, los signos de puntuación, paréntesis y otros signos similares, es necesario un conjunto de menos de 100 valores únicos. Por esta razón, las computadoras de unidad central de la "edad de piedra" (las décadas de 1960 y 1970) tenían como base lo que se conoce como codificación de 7 bits (dígitos binarios): con 7 bits se pueden codificar $2^7 = 128$ caracteres de manera unívoca. El estándar más conocido desarrollado sobre esta base es el llamado código ASCII (*American Standard Code for Information Interchange*, "Código estadounidense estándar para el intercambio de información"), véase la Tabla 1.

Tabla 1. Codificación estandarizada de 7 bits (ASCII)

	0										1									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
000																				
020														!	"	#	$	%	&	'
040	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;
060	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
080	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~													
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
						0										1				

Queda claro que con base en este esquema de codificación, los textos en inglés podrían ser digitalizados fácilmente, pero no los textos en alemán,

francés o español, y ni qué decir de los textos griegos, rusos o chinos en sus grafías originales. Sin embargo, esto no significa que en ese entonces fuera imposible procesar textos en lenguas "exóticas". Sólo había que inventar esquemas de codificación que utilizaran más de una unidad digital para representar ciertos caracteres. Véase en la Tabla 2 la adaptación a 7 bits de un texto en sánscrito, un pasaje del *Rigveda*, que se produjo en la década de 1970 en una computadora de unidad central, a la que se añade la trascripción "tradicional" para su comparación. Resulta claro que esta codificación tenía al menos dos desventajas: difícilmente era posible visualizar el texto en su forma original en una pantalla de computadora, lo que daba como resultado una gran cantidad de errores al introducir los datos, y la codificación no era transparente (no se "explicaba por sí misma"), en el sentido de que las unidades individuales (letras, diacríticos, signos de acentuación) fácilmente pudieron haber sido determinados por alguien que no estuviera involucrado en los procesos de codificación. Es cierto que esta codificación satisfacía la condición de ser consistente, ya que una determinada secuencia de códigos siempre representaba el mismo caracter, y esta es la razón de que estos textos se puedan utilizar y analizar incluso hoy en día. Sin embargo, resultaba demasiado torpe para ser sostenible durante un periodo más largo.

Tabla 2. Codificación no estandarizada de 7 bits (*Rigveda* 7, 1)

R700123011	AGNI!M+ NA!RO DI:!D)IT!B)IR ARA!N\YOR HA!STACYUT!: JANAYANTA PRAS=ASTA
R700123012	!M / DU:RED9!S=AM+ G9HA!PAT!M AT)ARYU!M
R700123021	TA!M AGNI!M A!STE VA!SAVO NY 9&N\VAN SUPRATICA!KS\AM A!VASE KU!TAS= CI
R700123022	T / DAKS\A:!YYO YO! DA!MA A:!SA NI!TYAH-
R700123031	PRE!DD)O AGNE DI:DIHI PURO! NO! 'JASRAYA: SU:RMYA:& YAVIS\T\)A / TVA:!
R700123032	M+ S=A!S=VANTA U!PA YANTI VA:!JA:H-
1	*agníṃ náro dídhitibhir aráṇyor hástacyutī janayanta práśastám / dūredṛ́śaṃ gṛhápatim atharyúm*
2	*tám agním áste vásavo ny ṛ̀ṇvan supraticákṣaṃ ávase kútaś cit / dakṣā́yyo yó dáma ā́sa nítyaḥ*
3	*préddho agne dīdihi puró nó 'jasrayā́ sūrmyā́ yaviṣṭha / tvā́ṃ śáśvanta úpa yanti vā́jāḥ*

1.2. PCs, Macs, DOS y MS Windows: estándares y no estándares basados en 8 bits

Este problema quedó superado al menos de manera parcial al extender a 8 bits la base de codificación ASCII. Con una base de 8 bits (= 1-byte) se pueden codificar de manera unívoca $2^8 = 256$ caracteres. Desde principios de la década de 1980 se desarrollaron y aplicaron muchos esquemas de codificación de 8 bits que añadían al inventario caracteres especiales como aquellos que representan las vocales con diéresis del alemán *ä, ö, ü* (con las que se representa la metafonía intervocálica), las vocales acentuadas *é, à, ô*, etc. del francés o la palatal nasal *ñ* del español. Desafortunadamente, esto no se hizo de una manera homogénea, "estandarizada", desde el principio; sino que algunas de las más importantes compañías de computadoras desarrollaron cada una su propio esquema. Esto provocó serios problemas cuando los datos habían de intercambiarse entre sistemas. Compárense las Tablas 3-5, que muestran los sistemas de codificación utilizados en las computadoras IBM/DOS, las computadoras Macintosh y el ambiente MS Windows. Sólo esta última es más o menos idéntica al estándar de 8 bits que hasta el día de hoy se utiliza en ambientes web, el estándar ANSI (*American National Standards Institute*, "Instituto nacional de estándares de Estados Unidos), también conocido como el estándar no. 8859-1 de la ISO (*International Standards Organization*, Organización Internacional para la Estandarización). Los caracteres especiales de MS-Windows se destacan con un fondo gris dentro de la Tabla 5.

Tabla 3. Codificación no estandarizada de 8 bits (sistema operativo DOS/IBM. ASCII Extendido, Página de códigos 437)

	0										1								
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
000		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼	►	◄	↕	‼
020	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼		!	"	#	$	%	&	'
040	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;
060	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
080	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~	⌂	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï
140	î	ì	Ä	Å	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	₧	ƒ
160	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»	░	▒	▓	│
180	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐	└	┴	┬	├	─	┼	╞	╟
200	╚	╔	╩	╦	╠	═	╬	╧	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█
220	▄	▌	▐	▀	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	∅	∈	∩
240	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■					
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
					0										1					

Tabla 4. Codificación no estandarizada de 8 bits (sistema operativo Mac OS de Macintosh)

	0										1								
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
000																				
020														!	"	#	$	%	&	'
040	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;
060	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
080	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~		Ä	Å	Ç	É	Ñ	Ö	Ü	á	à	â	ä	ã
140	å	ç	é	è	ê	ë	í	ì	î	ï	ñ	ó	ò	ô	ö	õ	ú	ù	û	ü
160	†	°	¢	£	§	•	¶	ß	®	©	™	´	¨	≠	Æ	Ø	∞	±	≤	≥
180	¥	µ	∂	Σ	Π	π	∫	ª	º	Ω	æ	ø	¿	¡	¬	√	ƒ	≈	Δ	«
200	»	…		À	Ã	Õ	Œ	œ	–	—	"	"	'	'	÷	◊	ÿ	Ÿ	⁄	¤
220	‹	›	fi	fl	‡	·	‚	„	‰	Â	Ê	Á	Ë	È	Í	Î	Ï	Ì	Ó	Ô
240		Ò	Ú	Û	Ù	ı	ˆ	˜	¯	˘	˙	˚	¸	˝	˛	ˇ				
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
					0										1					

Tabla 5. Codificación estandarizada de 8 bits (ANSI, ISO-8859-1, MS-Windows, Página de códigos 1252)

	0										1									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
000																				
020														!	"	#	$	%	&	'
040	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;
060	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
080	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~				,	ƒ	„	…	†	‡	ˆ	‰	Š	‹
140	Œ				‘	’	“	”	○	—	–	˜	™	š	›	œ				Ÿ
160		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯	°	±	²	³
180	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç
200	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û
220	Ü	Ý	Þ	ß	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ				
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
					0										1					

Aún así, estos sistemas no eran suficientes para la codificación inmediata de otras grafías, como las del griego, el cirílico o el chino. Por esto se desarrollaron desde mediados de la década de 1980 las llamadas "páginas de códigos" (*code pages*) para computadoras con base de 8 bits. Así como en los ejemplos anteriores, en estas páginas de códigos se utilizó el área "superior" (la de los valores superiores a 128), que excede el estándar ASCII básico, para codificar otros conjuntos de caracteres. Algunas de estas páginas de códigos se han estandarizado con la norma ISO-8859 (véase, por ejemplo, la Tabla 6, en la que se contrasta la página de códigos ISO-8859-5 para el cirílico con el estándar ANSI, que es la norma ISO-8859-1.

Tabla 6 a/b. Mapeo estandarizado de 8 bits: ISO-8859-1 (6a) vs. ISO-8859-5 (6b)

a.

32	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	47
48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	63
64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	79
80	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	95
96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	111
112	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~		127
160	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯			175
176	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿	191
192	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	207
208	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	223
224	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	239
240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	255

b.

32	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	47
48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	63
64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	79
80	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	95
96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	111
112	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~		127
160		Ё	Ђ	Ѓ	Є	S	І	Ї	J	Љ	Њ	Ћ	Ќ		Ў	Џ	175
176	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	191
192	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	207
208	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	223
224	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я	239
240	№	ё	ђ	ѓ	є	s	і	ї	j	љ	њ	ћ	ќ	§	ў	џ	255

Además de estas extensiones "oficiales", desde principios de la década de 1980 se desarrolló una cantidad desconocida de sistemas de codificación de 8 bits locales e incluso personales para satisfacer las necesidades de las lenguas y de los lingüistas. De hecho, cada vez que alguien desarrollaba o aplicaba cierto tipo de fuente tipográfica, cuya codificación no correspondiera con alguna de las páginas de códigos estandarizadas: se creaba un nuevo sistema de codificación desde cero. Después, al aplicar el método de "mapeo de caracteres" (*font mapping*), podíamos satisfacer, por ejemplo, los requisitos para anotar el griego antiguo (politónico) con sus caracteres originales o para representar las lenguas iranias con una trascripción latina (véanse las Tablas 7-8).

El problema de todo esto es que cuando se aplica el mapeo de caracteres no se pueden garantizar los requisitos básicos de documentación, es decir, la persistencia y la posibilidad de recuperación de los datos, porque no hay una correspondencia unívoca entre el caracter que ha de codificarse y un valor digitalizado asignado. Si, por ejemplo, aplicamos la fuente de 8 bits del griego ilustrada en la Tabla 7, el valor 231 representaría una letra griega *pi* (π) minúscula, mientras que el mismo valor representaría una *cha* (ч) cirílica si utilizáramos una fuente que equivaliera a la página de códigos estandarizada ISO-8859-5 y una *c* latina con cedilla (ç) si utilizáramos la norma ANSI básica. Esto significa que cada vez que se aplique una

codificación de 8 bits en la codificación de textos, se debe almacenar información adicional para indicar qué página de códigos o qué codificación de caracteres es válida para un caracter determinado. Sin embargo, esta información no se puede codificar como tal de manera estandarizada y se pierde fácilmente cuando los datos se transfieren de un sistema a otro. Un ejemplo será suficiente para ilustrar este fenómeno, que puede ser peligroso para el almacenamiento de textos a largo plazo.

Tabla 7. Codificación no estandarizada de 8 bits: griego antiguo (politónico)

	0										1									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
000		˙	˝	ˆ	˙	˝	ˆ	˙		˙										
020		§			˙	˙	˙	˙	˙	˙	ϟ	ϟ		!	"	῎	῏	῍	῾	'
040	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;
060	ή	ῆ	ὴ	?	ς	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
080	P	Q	R	S	T	U	V	W	X	Y	Z	[	ῃ	]	ῆ	˙	˙	a	b	c
100	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
120	x	y	z	ή	|	ή	ῆ	ᾰ	ᾰ	ῠ	ἐ	ᾰ	ᾱ	ὰ	ᾰ	ῐ	ῐ	ἔ	ὲ	ῑ
140	ῑ	ι	Α	ῐ	ό	ῠ	ἔ	ð	ὁ	ὸ	ῦ	ὺ	υ	Ὁ	Ὺ	ᾱ	ᾗ	ἰ	ὃ	ῦ
160	ά	ί	ό	ῦ	ὼ	ὼ	ὼ	ὼ	ὼ	ὼ	ὼ	ὼ	ἰ	ῦ	ᾰ	ή	ῆ	Γ	Δ	ῆ
180	ῆ	ῆ	Θ	ώ	ὼ	Λ	ὼ	ὼ	Ξ	ὼ	Π	ὼ	Σ	ὼ	ὼ	Φ	ὼ	Ψ	Ω	ᾳ
200	ί	ὺ	ᾳ	ᾳ	ή	ῆ	ῆ	ᾱ	ε	ί	ὁ	ὺ	ᾳ	ᾳ	ᾰ	α	ὼ	γ	δ	ε
220	ξ	η	ϑ	ι	κ	β	λ	μ	ν	ξ	ώ	π	ϱ	σ	τ	υ	φ	χ	ψ	ω
240	ϱ	ἰ	ὺ	ᾳ	ᾳ	η	ῳ	ᾳ	ά	ε	ι	ὁ	ύ	ῡ	ὅ					
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
					0										1					

Tabla 8. Codificación no estándar de 8 bits: fuente latina con diacríticos

	0											1							
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
000																				
020		§				Ł	Þ							!		#	†			
040	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;
060	<	=	>	?	√	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
080	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~	≈	ž	ü	é	â	ä	à	å	ç	ê	ë	è	ï
140	î	ì	Ä	o	é	æ	œ	ô	ö	ò	û	ù	ÿ	Ö	Ü	á	ē	ī	ó	ű
160	á	í	ó	ú	ñ	ŋ	ā	ē	ī	ō	ū	á	j	î	ł	ű	å	ě	ǐ	ı
180	ů	ą	ã	x	x^u	ž	$ŋ^u$	ţ	ī	ī	ú	ą	ę	į	ǫ	ų	į	ų	ə	ə
200	ə	ą	ą	ę	ę	ě	é	į	į	ų	ų	ű	ÿ	ý	β	b	c	d	đ	δ
220	ǵ	ĝ	g	ɣ	ḥ	ß	ḥ	ḥ	ḳ	ḷ	ḷ	ḷ	ḷ	ṃ	ṁ	ṁ	ṃ	ṅ	ń	ŋ
240	ṇ	ṛ	ṛ	ṛ	ṛ	ṛ	ś	ṣ	š	ŝ	ş	ṭ	ṭ	ð	þ					
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
						0										1				

1.3. Conversión y pérdida de datos: un ejemplo

En la Tabla 9a se muestran las diez primeras líneas de una canción popular en lengua suano (svan), digitalizada a principios de la década de 1980 en un ambiente DOS con una fuente especial que cubría los requisitos de la trascripción latina de las lenguas del sur del Caúcaso. Codificado como texto simple, sin ningún tipo de información adicional sobre la fuente, el texto habría aparecido como se muestra en la Tabla 9b en una pantalla de sistema DOS. Descifrar qué símbolo corresponde a la representación de qué caracter ciertamente sería una tarea difícil. Imaginemos a un lingüista que encontrara el texto dentro de 200 años y que no tuviera ninguna otra información sobre la lengua en cuestión (que para entonces quizás ya habría desaparecido, pues el suano pertenece a las "Lenguas caucásicas amenazadas de Georgia" del programa DoBeS).[1] Este lingüista no tendría

[1]Según el proyecto ECLinG (por sus siglas en inglés: Endangered Caucasian Languages in Georgia), que forma parte del programa DoBeS (*Dokumentation Bedrohter Sprachen*, "Documentación de Lenguas Amenazadas"). Véase la página web del proyecto en http://titus.fkidg1.uni-frankfurt.de/ecling/ecling.htm.

posibilidades de adivinar los valores de los caracteres cruciales y por lo tanto no podría recuperar el texto en sí mismo.

Tabla 9a/b. Mapeo de caracteres en una codificación de 8 bits: muestra de un texto en suano (svan)

	a.	b.
1	*voǯ ġal sabirelo Nuarsala!*	1 vo■ ╟al sabirelo Nuarsala!
2	*Muśvraši ṭubas esġəri.*	2 MuΩvraΩi ∩ubas es╟╨ri,
3	*sgobin laǯxvidax Ċolšare.*	3 sgobin la■xvidax ¬olΩare,
4	*min ǯixaldax si moḳtare,*	4 min ■ixaldax si mo╓tare,
5	*esran irix min amxvare.*	5 esran irix min amxvare.
6	*ka laǯšədax ečxän-amxän.*	6 ka la■Ω╨dax e─xaΩn-amxaΩn.
7	*meqrär šəqasuġv eǯlaǯix.*	7 meqraΩr Ω╨tasu╟v e■la■ix,
8	*ču laǯṭəxix Muśvra ṭubas.*	8 ─u la■∩╨xix MuΩvra ∩ubas.
9	*Davberxo lekva esqadäs,*	9 Davberxo lekva esqadäs,
10	*Davbrar q̇ǫ̈rars xocqanalix:*	10 Davbrar ∞rare xocqanalix:
11	*lənšare sgoǯix muśgvriša.*	11 l╨mΩare sgo■ix muΩgvriΩa.

1.4. Unicode: hacia un estándar mundial

¿Cuál es, pues, la solución a este problema? La respuesta es clara: para codificar de manera unívoca todos los caracteres que se han utilizado para escribir las lenguas de la humanidad (incluyendo desde grafías y alfabetos "nacionales" hasta "metagrafías" lingüísticas, como el Alfabeto Fonético Internacional, IPA por sus siglas en inglés: *International Phonetic Alphabet*),.la base de codificación debe extenderse mucho más allá del estándar de 1 byte (8 bits). Esto es exactamente lo que se intentó desde principios de la década de 1990 cuando se creó el estándar *Unicode*: con su base de 16 bits (o 2 bytes), esta norma comprende 2^{16} = 65,536 combinaciones utilizadas para la codificación "unívoca" de caracteres. Si se considera que tan sólo para la grafía del chino se han utilizado más de 65,000 caracteres diferentes a través de la historia, resulta claro que incluso este estándar no es todavía suficiente para cubrir todos los caracteres que la humanidad ha utilizado en todos los tiempos. Sin embargo, se está desarrollando una extensión adicional mediante la norma ISO 10646 de 32

bits, que genera un total de 2^{32} = 4,294,967,296 puntos de código. De hecho, el estándar Unicode no es sino un subconjunto de este inventario "infinito", así como el estándar ANSI (ISO 8859-1) es un subconjunto de Unicode y el estándar ASCII es un subconjunto del ANSI (véase la Figura 1).

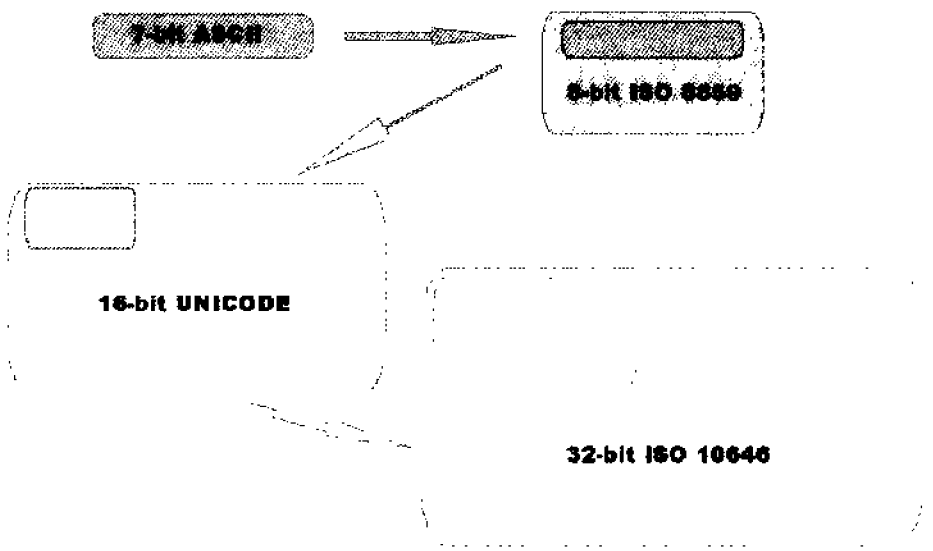


Figura 1. De la codificación de 8 bits a la de 32 bits

El estándar Unicode ha cobrado una importancia creciente desde finales de la década de 1990, a la par de la expansión de la red mundial *World Wide Web*, y ahora es la base de la codificación en los sistemas operativos y procesadores de texto más actualizados. Sin duda, esto representa una enorme ventaja para los propósitos de la documentación lingüística. Véanse, por ejemplo, las Tablas 10a y 10b, en las que se muestran algunos de los "bloques" de caracteres Unicode: ahora queda garantizada la distinción entre una *che* (ч) cirílica y una *c* latina con cedilla (ç) porque sus códigos están diferenciados (número hexadecimal 0447 = decimal 1095 *vs.* hexadecimal 00E7 = decimal 231). Además, ahora muchos caracteres de base latina utilizados en sistemas de trascripción se pueden codificar como caracteres griegos, georgianos o chinos.

Tabla 10 a/b. Codificación en 16 bits: bloques Unicode para los alfabetos latino (a) y cirílico (b)

a.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000																
001																
002	?	!	"	#	$	%	&	'	(	)	*	+	,	-	.	
003	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
006	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
008																
009																
00A		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
00B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
00C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
00D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

b.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
040	Ѐ	Ё	Ђ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	Ѝ	Ў	Џ
041	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
042	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
043	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
044	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
045	ѐ	ё	ђ	ѓ	є	ѕ	і	ї	ј	љ	њ	ћ	ќ	ѝ	ў	џ
046	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ
047	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѻ	Ѽ	ѽ	Ѿ	ѿ

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
048	Ҁ	ҁ	҂	҃	҄	҅	҆	҇	҈	҉	Ҋ	ҋ	Ҍ	ҍ	Ҏ	ҏ
049	Ґ	ґ	Ғ	ғ	Ҕ	ҕ	Ж	ж	Ҙ	ҙ	Қ	қ	Ҝ	ҝ	Ҟ	ҟ
04A	Ҡ	ҡ	Ң	ң	Ҥ	ҥ	Ҧ	ҧ	Ҩ	ҩ	Ҫ	ҫ	Ҭ	ҭ	Ү	ү
04B	Ұ	ұ	Ҳ	ҳ	Ҵ	ҵ	Ҷ	ҷ	Ҹ	ҹ	Ҽ	ҽ	Ҿ	ҿ	Ӏ	Ӂ
04C	Ӄ	ӄ	Ӆ	ӆ	Ӈ	ӈ	Ӊ	ӊ	Ӌ	ӌ	Ӎ	ӎ	Ӑ	ӑ	Ӓ	ӓ
04D	Ӕ	ӕ	Ӗ	ӗ	Ә	ә	Ӛ	ӛ	Ӝ	ӝ	Ӟ	ӟ	Ӡ	ӡ	Ӣ	ӣ
04E	Ӥ	ӥ	Ӧ	ӧ	Ө	ө	Ӫ	ӫ	Ӭ	ӭ	Ӯ	ӯ	Ӱ	ӱ	Ӳ	ӳ
04F	Ӵ	ӵ	Ӷ	ӷ	Ӹ	ӹ	Ӻ	ӻ	Ӽ	ӽ	Ӿ	ӿ				

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Quizás valga la pena destacar que Unicode no fue el primer intento por prevenir el caos de los mapeos de caracteres de 8 bits en codificaciones de 16 bits. En 1988 apareció el procesador de palabras WordPerfect 5.0 (WP 5), que contenía un conjunto de 1632 caracteres codificados de manera unívoca, incluidos conjuntos de griego, cirílico y japonés (*hiragana* y *katakana*) además de un bloque de 255 entidades que el propio usuario podía definir (la llamada *user definable area*). De esta manera, los textos codificados en WP 5 pueden cumplir con los requisitos de codificación unívoca de caracteres incluso hoy día y es posible mantener intacta la información que contienen cuando estos textos se transfirieran al estándar Unicode. Desafortunadamente, el sistema de codificación de WP no tuvo un uso tan generalizado y se pasaron por alto las oportunidades que ofrecía; así, no podemos esperar que las rutinas de conversión automática para textos de WP 5 que ofrece, por ejemplo, MS Word 2000 interpreten correctamente y recodifiquen cualquiera de los caracteres no estandarizados

que pudieran estar incluidos en esos textos. En la Tabla 11 se ilustra lo que pasa cuando la canción popular en suano mencionada antes (véase Tabla 9) se codifica de manera consistente en WP 5 y después se convierte automáticamente en texto de MS Word. En particular, la versión convertida queda inutilizable e irreparable porque los caracteres "no identificados" son remplazados por un guión bajo no diferenciado (_). Lo mismo ocurre con la conversión automática proporcionada por versiones posteriores del mismo WordPerfect (por ejemplo WP 9); en ésta, por ejemplo, la ə se reemplazó por una B, que es obviamente confuso. Todo esto significa que una conversión correcta de textos codificados en WordPerfect 5 (o por lo menos de los caracteres en ellos contenidos) a codificación Unicode es posible, pero requiere de una programación especial.

Tabla 11 a/b. "Conversión" automática de texto: ejemplo del suano (svan)

	a.		b.
1	*vo_ Ÿal sabirelo Nuarsala!*	1	*voÁ ġal sabirelo Nuarsala!*
2	*Mušvraši _ubas esŸ_ri,*	2	*Mušvraši Mubas esġBri,*
3	*sgobin la_xvidax _olšare,*	3	*sgobin laÁxvidax volšare,*
4	*min _ixaldax si mo_tare,*	4	*min Áixaldax si moåtare,*
5	*esran irix min amxvare.*	5	*esran irix min amxvare.*
6	*ka la_š_dax e...xän-amxän,*	6	*ka laÁšBdax ecxän-amxän,*
7	*meqrär š__asuŸv e_la_ix.*	7	*meqrär šBïasuġv eÁlaÁix,*
8	*...u la__xix Mušvra _ubas.*	8	*ču laÁMBxix Mušvra Mubas.*
9	*Davberxo lekva es_adäs,*	9	*Davberxo lekva esïadäs,*
10	*Davbrar _Çvars xoc_analix:*	10	*Davbrar ïõrars xocïanalix:*
11	*l_mšare sgo_ix mušgvriša.*	11	*lBmšare sgoÁix mušgvriša.*

1.5. Estándares coexistentes: el peor de los casos

Ahora la pregunta es: ¿realmente estamos en un terreno seguro después de que Unicode se convirtió en la norma mundial para la codificación de caracteres? Hay que reconocer que todavía hay muchos problemas complicados sin resolver, y no sólo con respecto a la conversión de

materiales antiguos. El principal problema está en el hecho de que, por el momento, el procesamiento digital de palabras se caracteriza por la coexistencia real de sistemas de codificación de 16 bits y 8 bits. Así como el estándar ANSI de 8 bits se incorporó al estándar Unicode de 16 bits como uno de sus "bloques", todos los procesadores de palabras basados en Unicode, como MS Word 2000, se han diseñado para poder manejar textos codificados tanto en 8 bits como en 16 bits. De la misma manera, los sistemas operativos basados en Unicode, como MS Windows 2000, se han diseñado para incorporar fuentes tipográficas codificadas en 8 bits junto a las codificadas en 16 bits. Unos cuantos ejemplos bastarán para demostrar la confusión que esto puede causar.

La Tabla 12 muestra una lista de verbos georgianos capturada en MS Word 6, utilizando una fuente georgiana simple de 8 bits mapeada según el esquema de codificación ANSI de 8 bits. Cuando recibí este archivo de texto de un colega en Georgia, vía correo electrónico, hace dos años, traté de abrirlo en MS Word 2002 (XP Office). El resultado fue extraño, por decir lo menos: en lugar de su texto, apareció en pantalla un texto en la escritura japonesa *katakana* (véase la Tabla 12b). Cuando abrí el texto en Open Office 1, obtuve otro resultado: los caracteres georgianos habían sido remplazados por caracteres latinos con diacríticos (véase la Tabla 12c), un resultado previsible si se toma en cuenta que la codificación original estaba basada en 8 bits. Tras aplicar la fuente georgiana correcta a este texto dentro de Open Office, reapareció el aspecto buscado (como en la Tabla 12a) y el texto pudo ser re-mapeado a una fuente de trascripción que utilizaba los mismos puntos de código de 8 bits (véase la Tabla 12d). Tratar de aplicar la fuente georgiana a los caracteres japoneses que aparecieron en pantalla con MS Word 2002 no cambió nada, pues los caracteres *katakana* siguieron siendo caracteres *katakana* (como se muestra en la Tabla 12b).

Tabla 12 a-d. "Conversión" automática de texto: ejemplo del georgiano (lista de palabras)

a. Texto original (MS Word 6)

0020010M	გააღვილება (გაადვილებ-ისა)	0020020M	გააშნაურება (გააშნაურებ-ისა)
0020030M	გაბმა (გაბმ-ისა)	0020040N	გაგა-ჲ (გაგ-ისა)
0020050M	გაგება (გაგებ-ისა)	0020060P	გაგებულ-ი (გაგებულ-ისა)
0020070M	გაგზავნა (გაგზავნ-ისა)	0020080N	გაგზავნა-ჲ (გაგზავნ-ისა)

b. El mismo texto después de una transferencia entre versiones de un mismo programa (MS Word 6 > MS Word 2002)

0020010M ッタタテナノヒトチタ (ッタタテナノヒトチ - ノモタ)
0020020M ッタタニへタヨメトチタ (ッタタニへタヨメトチ - ノモタ)
0020030M ッタチフタ (ッタチフ - ノモタ)
0020040N ッタツタ - ホ (ッタツ - ノモタ)
0020050M ッタツトチタ (ッタツトチ - ノモタ)
0020060P ッタツトチヨヒ - ノ (ッタツトチヨヒ - ノモタ)
0020070M ッタツニタナへタ (ッタツニタナへ - ノモタ)
0020080N ッタツニタナへタ - ホ (ッタツニタナへ - ノモタ)

c. El mismo texto después de transferencia entre programas (MS Word 6 > Open Office 1)

0020010M ÅÀÀÃÅÉËÄÁÀ (ÅÀÀÃÅÉËÄÁ-ÉÓÀ)
0020020M ÅÀÀÆÍÀÖÒÄÁÀ (ÅÀÀÆÍÀÖÒÄÁ-ÉÓÀ)
0020030M ÅÀÁÍÀ (ÅÀÁÍ-ÉÓÀ)
0020040N ÅÀÅÀ-Î (ÅÀÀ-ÉÓÀ)
0020050M ÅÀÂÄÁÀ (ÅÀÂÄÁ-ÉÓÀ)
0020060P ÅÀÀÄÀÖË-É (ÅÁÁÄÁÖË-ÉÓÀ)
0020070M ÅÀÅÆÀÁÍÀ (ÅÀÂÆÀÁÍ-ÉÓÀ)
0020080N ÂÀÀÆÁÁÍÁ-Î (ÅÁÁÆÁÁÍ-ÉÓÀ)

d. Lo mismo ocurre al aplicar una fuente diferente (dentro de Open Office 1)

0020010M gaadvileba (gaadvileb-isa)
0020020M gaaznaureba (gaaznaureb-isa)
0020030M gabma (gabm-isa)
0020040N gaga-j (gag-isa)
0020050M gageba (gageb-isa)
0020060P gagebul-i (gagebul-isa)
0020070M gagzavna (gagzavn-isa)
0020080N gagzavna-j (gagzavn-isa)

¿Cómo puede explicarse esta conducta extraña de MS Word? Obviamente, el programa ejecuta una estrategia de cinco pasos cuando se encuentra con textos codificados en otras versiones (más antiguas):

1) Verifica si el documento está codificado en Unicode.
2) En caso negativo, verifica si la distribución de caracteres coincide con la distribución "típica" de alguna de las páginas de códigos conocidas.
3) En caso afirmativo, da por hecho que esa página de códigos es la que debe representarse.
4) Convierte los caracteres de 8 bits de la página de códigos aceptada como correcta a los caracteres equivalentes de Unicode.

5) Almacena los caracteres Unicode en la memoria.

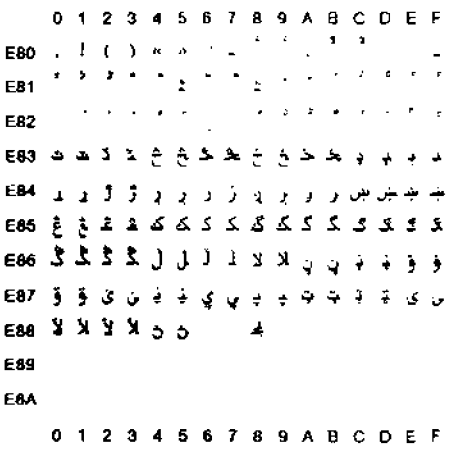
Entonces, puede ser que volver a aplicar las fuentes de 8 bits originales no sea un remedio si éstas no satisfacen la codificación que se aceptó como Unicode, como en el ejemplo dado, en el que se dio por hecho que se trataba de una página de códigos en japonés.

1.6. No estándares persistentes: el "Área de Uso Privado"

Otro problema que podría ser crucial incluso en tiempos de Unicode es la persistencia de por lo menos un área que está diseñada para el mapeo personalizado de fuentes. Es la llamada "Área de Uso Privado" (PUA por sus siglas en inglés: *Private Use Area*) que abarca 6144 caracteres no predefinidos en los bloques E000-EFFF y F000-F7FF. Similar al área definible por el usuario de WordPerfect 5, la PUA puede ser asignada *ad libitum* por compañías, grupos de usuarios o individuos. Esto da como resultado que se necesite información adicional para distinguir los caracteres que se "codificaron" en ella. En la Tabla 13 se muestra lo que podría pasar cuando se aplica una fuente equivocada para visualizar caracteres codificados en la PUA: en el peor de los casos, la información prevista se perderá otra vez.

Tabla 13 a/b. Mapeo de fuentes de 16 bits: el "Área de Uso Privado"

a.



b.

controladores de teclado especiales que hayan sido proporcionados por terceros, tales como Tavultesoft Keyman, ya que pueden haber sido diseñados únicamente para codificaciones de 8 bits y esto anularía las posibilidades de insertar con ellos textos codificados en 16 bits. Si lo que se pretende es diseñar un controlador de teclado propio con Keyman o con MS Keybord Layout Creator, habrá que asegurarse de utilizar la codificación Unicode como base. Por cierto, el programa SIL Shoebox estaba basado exclusivamente en 8 bits e interactuaba bien con los controladores Keyman, pero únicamente con base de 8 bits. La nueva versión, Toolbox, tiene base Unicode y debería trabajar bien con los diseños de teclado Keyman con base de 16 bits.

2. La codificación de elementos textuales: Apariencia superficial *vs.* marcado de contenido

2.1. La estructura textual visualizada

Tratemos ahora el segundo tema de este capítulo, es decir, la codificación de los elementos estructurales de los textos. Para aclarar lo que esto significa, será útil volver a ver el texto en suano con el que hemos trabajado antes (véase la Tabla 9). Incluso sin ningún conocimiento de la lengua, de inmediato tenemos la impresión de que este texto está formado por versos. Esto está claramente indicado por dos señales a las que estamos acostumbrados en la lectura de textos poéticos, a saber: las líneas son relativamente cortas, y están enumeradas (de la 1 a la 11). Sin embargo, hay muchos más elementos de estructura textual involucrados. Primero, es fácil suponer que el texto consta de cinco oraciones, que se extienden de manera parcial entre los versos y que consisten, en parte, de cláusulas subordinadas: esto está indicado por los signos de puntuación. Después, podemos afirmar que el texto consta de 38 palabras, a su vez indicadas ya sea por espacios en blanco o por signos de puntuación colindantes a los primeros o últimos caracteres.

2.1.1. Los elementos básicos

Para la documentación de materiales textuales, aunque pueda parecer trivial, es determinante considerar y marcar los elementos internos cuando se les prepara para el uso futuro y esto deberá hacerse de manera tan consistente

como la codificación de los caracteres que aparecen en las palabras. ¿De qué elementos estamos hablando? Entre los elementos básicos de todo tipo de texto, ya hemos mencionado las palabras (que en su forma escrita están compuestas de caracteres), frases, cláusulas y oraciones; a un nivel superior, encontramos secciones, párrafos, capítulos, partes de texto y cosas por el estilo. Para muchos de estos elementos, intuitivamente adaptamos señales a las que estamos acostumbrados desde que íbamos a la escuela, como los espacios que señalan la división de palabras, los puntos finales que indican el final de una oración o el nuevo renglón que indica el inicio de una sección o de un párrafo. Sin embargo, esto podría no ser suficiente para la codificación consistente de un texto digital. Otro ejemplo bastará para demostrar por qué no.

2.1.2. Un ejemplo ilustrativo

En la Tabla 14 vemos un extracto de un tratado gramatical en georgiano del siglo XVIII digitalizado con MS Word 6. Sin tener el mínimo conocimiento de la escritura georgiana, un lector podría suponer que la primera línea del texto es un encabezado o un título, puesto que, obviamente, consiste tan sólo de una palabra, está al centro de la línea y parece estar representada en negritas. En lo que se refiere a las otras líneas de texto, el lector fácilmente sospechará que se trata de una interacción de preguntas y respuestas, esto está claramente indicado por los signos de interrogación. Otra sugerencia podría imponerse: como la primera palabra de cada pregunta y respuesta está separada por dos puntos y destacada por un espaciado adicional entre caracteres, y como estas palabras se repiten a través de las preguntas y respuestas, podrían ser los nombres de las personas que hablan (como en una obra de teatro). Todas estas suposiciones son correctas: tenemos una interacción de preguntas y respuestas enunciadas aquí por dos personas diferentes (una es Ioane, la otra, Nikolaoz) y la primera línea es el título (que significa simplemente "Sobre la gramática"). La razón por la que fue tan fácil descubrir todo esto es que en este caso, una vez más, se utilizaron métodos de marcado a los que estamos acostumbrados al leer: centrado de líneas, uso de negritas, espaciado entre caracteres, etc. Sin embargo, para propósitos computacionales, estas marcas, a las que llamaremos "orientadas hacia la superficie" (*surface-oriented*), son arbitrarias e insuficientes en dos sentidos.

Tabla 14. Muestra de texto georgiano

ღრამმატიკისათჳს

იოანემ: ოთხნი იგი გვარნი მოძღვრებითნი, რომელნიცა შეუდგებიან, დაემღვრებიან ღრამმატიკასა.

ნიკოლაოზმან: რაჲ არს სახელები მათი?

იოანემ: განსაზღვრება, განწეალება, აღმოჩენა და აღლეგა.

ნიკოლაოზმან: კვალად რაჲ საჭმარ არს ცნობად?

2.1.3. Características de programa vs. estándares

Primero, el centrado de las líneas puede ser una característica común de todos los procesadores de palabras que existen hoy en día, pero de ninguna manera está estandarizado: la codificación de esta característica simplemente depende de la estructura del programa. Para ilustrar lo que esto significa, la Tabla 15 muestra una parte del código interno del texto georgiano en MS Word. Aquí podemos localizar la palabra incluida en el encabezado (ღრამმატიკისათჳს, "Sobre Gramática" en georgiano, almacenada en forma de 8 bits) al final de lo que parece ser una sexta línea, seguido de las preguntas y respuestas en forma de "texto legible". No hay ninguna indicación colindante a la palabra que corresponde al título de que ésta deba estar centrada o en negritas, ni tampoco que represente un encabezado. Todo esto lo debe inferir el programa que lo interpreta, partiendo del código ilegible que lo precede (o de un bloque de elementos de codificación similar que se añade al final de cada documento de MS Word). Imaginemos que alguien tuviera que decodificar este documento dentro de 200 años, sin tener ningún acceso a la estructura de códigos interna del programa MS Word 6; ciertamente, esta persona no sería capaz de extraer nada salvo el "texto simple", y toda la información adicional referente al centrado de líneas y el texto en negritas se perdería (de hecho, muchos de nosotros hemos experimentado esto cuando tratamos de abrir documentos de MS Word de la década de 1980 en versiones posteriores). Lo mismo ocurriría con los caracteres "espaciados" que indican a los hablantes en el texto. Este espaciado también está cubierto por una función interna del programa y se perdería junto con el conocimiento del código.

Por cierto, no sería buena idea evitar este efecto al insertar manualmente los espacios entre caracteres en vez de usar la función específica para ello del procesador de palabras, pues como utilizamos espacios para distinguir unas palabras de otras, cualquier análisis informático leería el nombre espaciado I O A N E como cinco palabras (cada una formada por un caracter) y no lo encontraría si se buscara como "IOANE".

Tabla 15. Muestra del texto en georgiano en codificación específica de programa

```
Ðïà¡±ÿáÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿAÿþÿCÿÿÿÿÿÿÿÿÿÿÿÿAÿÿÿÿÿ´Aÿÿÿÿÿ0
ÿÿÿÿÿÿÿ BÿÿÿÿÿÀÿÿÿÿÿÿFÿÿÿMicrosoft Word 6.0 Document ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿMSWord-
DocÿÿÿÿWord.Document.6ÿô9²qÿÿÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
ÿÿÿ  Aÿÿÿÿÿ´Aÿÿÿÿÿ0Aÿÿÿ  Aÿÿÿÿÿ´Aÿÿÿÿÿ0AÿC:\WORKDIR\TEMPLATE\
NORMAL.DOTÿÿ
ÿÿÿüAÿÿÿÿÿ BÿÿÿÿÿDBÿÿ ÿÿÿhBÿÿÿÿÿŒBÿÿÿÿÿÿÿÿÿÿÙRAMMAºI±ISATWS
IOANEM: OTXNI IGI GVARNI MO¾ÙVREBITNI, ROMELNICA ¸EUdGEBIAN,
dAEMdEVREBIAN ÙRAMMAºI±ASA.
NI±OLAOzMAN: RAJ ARS SAXELEBI MATI?
IOANEM: GANSAzÙVREBA, GANÂVALEBA, AÙMOÆENA dA AÙLEVA.
NI±OLAOzMAN: ±VALAd RAJ SAQMAR ARS CNOBAd?
```

2.1.4. Lo que se ve NO es lo que se obtiene

Entonces, ¿qué se puede hacer para evitar la pérdida de información referente a la estructura de los textos y a sus elementos? Primero, debemos deshacernos de un ideal dentro del procesamiento de textos, que se ha generalizado en nuestros días, a saber: "Lo que se ve es lo que se obtiene" (en inglés se usan las siglas WYSIWYG: *What you see is what you get*). Puede ser cierto que el texto que se teclee en la computadora hoy se vea igual en la pantalla de la computadora o en una impresión, pero todo esto queda restringido a un uso efímero: la próxima generación de usuarios del texto quizás no tenga acceso a las sofisticadas codificaciones del procesador de palabras utilizado para capturarlo y por lo tanto obtendrá cualquier cosa salvo "lo que se vio". Segundo, debemos renunciar a la idea de que los mecanismos tipográficos (como negritas, el espaciado de caracteres y otros por el estilo) pudieran ser suficientes para indicar la función de los elementos del texto. En vez de eso, debemos adaptarnos a lo que se podría

llamar "marcado de contenido" (*content markup*) en caso de que los textos se almacenen para propósitos de documentación.

2.2. Una solución intermedia: HTML

En años recientes, el marcado de los elementos textuales se ha generalizado cada vez más, en especial con la expansión de la red mundial *World Wide Web* y con la necesidad de usar determinado tipo de estructura de codificación de texto unificada para documentos que han de subirse a la red. Esta estructura se llama HTML (*HyperText Markup Language*, Lenguaje de Marcado de Hipertexto). Las Tablas 16a y 16b presentan una muestra de texto en georgiano convertido a HTML (como código fuente y visualizado con un navegador web estándar); aquí se encontrarán fácilmente los dispositivos de marcado correspondientes al centrado y a las negritas del encabezado, es decir, los marcadores ... y ... . Lo que no se encontrará es el marcado especial de los nombres de los hablantes, porque el espaciado entre caracteres no puede marcarse como tal en HTML. Aunque pueden usarse para este marcado las llamadas "hojas de estilo en cascada" (*cascading style sheets, CSS*), no sería buena idea recurrir sólo a estas hojas, porque como el espaciado de caracteres no tiene un significado estandarizado, los futuros usuarios difícilmente tendrían idea de qué representa. De la misma manera, sigue siendo poco claro qué indican el centrado y las negritas de la primera línea: que se trata de un encabezado es una mera suposición. De hecho, el marcado que HTML proporciona contiene muy pocos elementos "de contenido". Uno es el grupo de marcadores de <H1> a <H6>, que debería utilizarse para indicar varios niveles de encabezado. En nuestro caso, sería mucho mejor marcar nuestro encabezado con uno de estos elementos (remplazando ... por <h1 align=center> ... </h1>). Así, la apariencia exterior sería secundaria y adaptable a usos futuros.

Tabla 16a. Codificación en HTML simple de la muestra de texto georgiano

```
<HTML>
    <HEAD>
        <META HTTP-EQUIV="Content-Type" CONTENT="text/html;
        charset=iso-8859-1">
        <TITLE>Grammatika</TITLE>
        <META NAME="KeyWords" CONTENT="Georgian Grammar">
            <BODY>
                <DIV>
                    <P ALIGN="CENTER"><B>ÙRAMMA°I±ISATWS</B></P>
                </DIV>
                <DIV>
                    <P><SPAN>IOANEM: </SPAN><SPAN>OTXNI IGI GVARNI
                    MO¾ÙVREBITNI, ROMELNICA ¸EUdGEBIAN, dAEMdEVREBIAN
                    ÙRAMMA°I±ASA.</SPAN></P>
                    <P><SPAN>NI±OLAOzMAN: </SPAN><SPAN>RAJ ARS SAXELEBI
                    MATI?</SPAN></P>
                    <P><SPAN>IOANEM: </SPAN><SPAN>GANSAzÙVREBA,
                    GANÂVALEBA, AÙMOÆENA dA AÙLEVA.</SPAN></P>
                    <P><SPAN>NI±OLAOzMAN: </SPAN>±VALAd RAJ SAQMAR ARS
                    CNOBAd?</SPAN></P>
                    ...
                </DIV>
            </BODY>
</HTML>
```

Tabla 16b. Apariencia de la muestra de texto georgiano en HTML vista en un navegador

ÙRAMMA°I±ISATWS

IOANEM: OTXNI IGI GVARNI MOÙVREBITNI, ROMELNICA ¸EUdGEBIAN, dAEMdEVREBIAN ÙRAMMA°I±ASA.

NIKOLAOZMAN: RA ARS SAXELEBI MATI?

IOANEM: GANSAzÙVREBA, GANÂVALEBA, AÙMOÆENA DA AÙLEVA.

NIKOLAOZMAN: ±VALAD RA SAQMAR ARS CNOBAD?