1 Juegos de caracteres

1.1 ASCII

El código ASCII, acrónimo inglés de American Standard Code for Information Interchange (Código Estadounidense Estándar para el Intercambio de Información), fue creado en 1963 por el Comité Estadounidense de Estándares. Más tarde, en 1967, se incluyeron las minúsculas, y se redefinieron algunos códigos de control para formar el código conocido como US-ASCII.

El código ASCII utiliza 7 bits para representar los caracteres, aunque inicialmente empleaba un bit adicional (bit de paridad) que se usaba para detectar errores en la transmisión.

Los códigos de 0 al 31 no se utilizan para caracteres imprimibles y se denominan *caracteres de control*.

Los códigos 65 al 90 representan las letras mayúsculas y los códigos 97 al 122 las letras minúsculas. Si cambiamos el 6º bit, se pasa de mayúscula a minúscula; esto equivale a agregar 32 al código ASCII en base decimal.

La tabla de códigos ASCII es la que se muestra a continuación:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
00	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
10	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
20		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	E	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	U	v	w	x	y	z	{	|	}	~	DEL

Nota: El carácter 0x20 es el espacio

1.2 ASCII extendido

El ASCII se desarrolló para utilizarse con el idioma inglés y no posee caracteres acentuados, o caracteres específicos de otros idiomas. Para codificar estos caracteres, se necesitaba un sistema de códigos distinto. Así, el código ASCII se extendió a 8 bits, denominándose código ASCII extendido. Este juego de caracteres es igual que el ASCII original en los 128 primeros códigos.

El código ASCII extendido no está estandarizado y varía de acuerdo a la plataforma en que se utiliza. Los dos grupos de caracteres más comunes del código ASCII extendido son:

→ El código extendido ASCII OEM, que estaba integrado en el primer PC de IBM, también conocido como OEM y de forma genérica como ASCII extendido o simplemente ASCII y cuya tabla se muestra a continuación:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	₧	ƒ
A	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	ø	ε	∩
F	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

→ El código extendido ASCII ANSI, utilizado por los sistemas operativos Windows, también conocido como ANSI y cuya tabla se muestra a continuación:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	□	□	‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ	□	□	□
9	□	‘	’	“	”	•	–	—	˜	™	š	›	œ	□	□	Ÿ
A		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

1.3 EBCDIC

El código EBCDIC (en castellano, código de intercambio decimal binario extendido), desarrollado por IBM, se utiliza para codificar caracteres con 8 bits. A pesar de que IBM lo utiliza en muchos de sus equipos, no ha tenido tanto éxito como ASCII.

1.4 Unicode

Unicode es un sistema de codificación de caracteres de 16 bits desarrollado en 1991. Unicode puede representar cualquier carácter a través de un código de 16 bits, independientemente del sistema operativo o el idioma de programación utilizado.

Incluye casi todos los alfabetos actuales (como el árabe, el armenio, el cirílico, el griego, el hebreo y el latín) y es compatible con el código ASCII.

Encontrará una lista de todos los códigos que se utilizan en Unicode en http://www.unicode.org.

1.5 Unicode Transformation Format-8

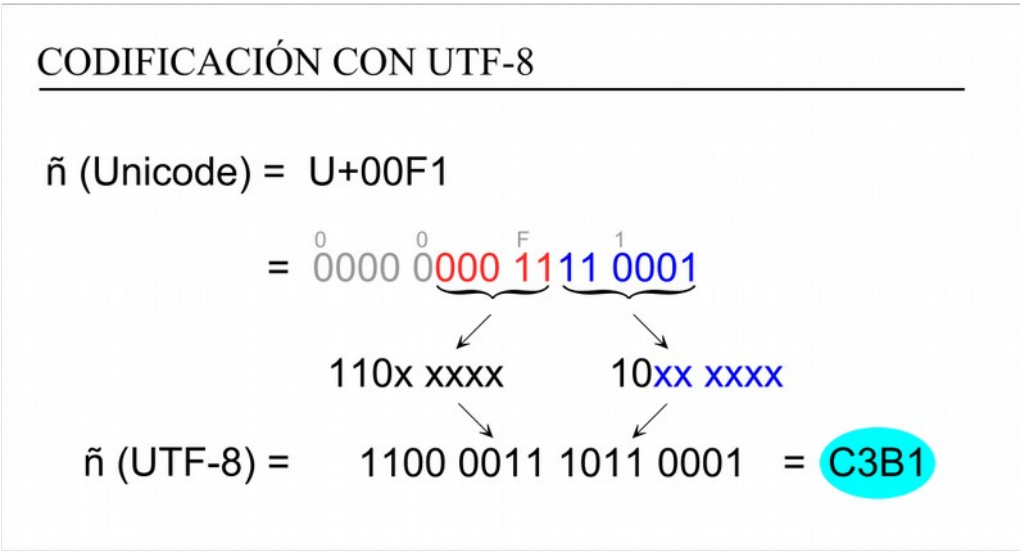
Unicode es el estándar que albergaba todas las lenguas de la tierra, sin excepciones, pero ¿cómo representamos Unicode para no provocar problemas en nuestros sistemas? Mediante la codificación UTF-8, cuyas premisas son:

- → El bit más significativo de un carácter de byte-simple es siempre 0
- → Los bits más significativos del primer byte de una secuencia multi-byte determinan la longitud de la secuencia. Estos bits más significativos 110 para secuencias de dos bytes; 1110 para secuencias de tres bytes, etc.
- → Los bytes restantes en una secuencia multi-byte tienen 10 como sus 2 bits más significativos.

De los puntos anteriores se genera la siguiente tabla de transformación:

Rango Unicode (hexadecimal)		UTF-8 secuencia de octetos (binario)
0000 0000	0000 007F	0xxxxxxx
0000 0080	0000 07FF	110xxxxx 10xxxxxx
0000 0800	0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
0001 0000	0010 FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
0020 0000	03FF FFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
0400 0000	7FFF FFFF	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

A continuación se muestra un ejemplo de codificación del carácter Unicode correspondiente a la letra ñ en UTF-8:



UTF-8 es una de las formas de representar Unicode, pero existen otras alternativas como UTF-16, UTF-32, UTF-7, SCSU.