

# Trabajo práctico N° 6

## Representación digital de datos: Texto y Multimedia

FECHA DE FINALIZACIÓN: 17 DE MAYO



Introducción a la computación  
Departamento de Ingeniería de Computadoras  
Facultad de Informática - Universidad Nacional del Comahue



**Objetivo:** comprender la representación binaria de texto, imágenes y otros datos más complejos.

### Recursos web:

- Wikipedia: *Run-length encoding*: [https://en.wikipedia.org/wiki/Run-length\\_encoding](https://en.wikipedia.org/wiki/Run-length_encoding)
- Wikipedia: *ASCII*: <http://es.wikipedia.org/wiki/ASCII>
- Tabla de caracteres ASCII Extendida: <http://www.programasprogramacion.com/caracteres.php>
- Tabla de caracteres *UTF-8*: <http://www.fileformat.info/info/charset/UTF-8/list.htm>

### Lectura obligatoria:

- Apuntes de cátedra. Capítulo 4: Representación digital de datos: Texto y Multimedia. Disponible en: <https://egrosclaude.github.io/IC/IC-notes.pdf>

### Tabla *ASCII*:

Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex										
0	00	NUL	16	10	DLE	32	20	48	30	0	64	40	@	80	50	P	96	60	'	112	70	p	
1	01	SOH	17	11	DC1	33	21	!	49	31	1	65	41	A	81	51	Q	97	61	a	113	71	q
2	02	STX	18	12	DC2	34	22	"	50	32	2	66	42	B	82	52	R	98	62	b	114	72	r
3	03	ETX	19	13	DC3	35	23	#	51	33	3	67	43	C	83	53	S	99	63	c	115	73	s
4	04	EOT	20	14	DC4	36	24	\$	52	34	4	68	44	D	84	54	T	100	64	d	116	74	t
5	05	ENQ	21	15	NAK	37	25	%	53	35	5	69	45	E	85	55	U	101	65	e	117	75	u
6	06	ACK	22	16	SYN	38	26	&	54	36	6	70	46	F	86	56	V	102	66	f	118	76	v
7	07	BEL	23	17	ETB	39	27	'	55	37	7	71	47	G	87	57	W	103	67	g	119	77	w
8	08	BS	24	18	CAN	40	28	(	56	38	8	72	48	H	88	58	X	104	68	h	120	78	x
9	09	HT	25	19	EM	41	29	)	57	39	9	73	49	I	89	59	Y	105	69	i	121	79	y
10	0A	LF	26	1A	SUB	42	2A	*	58	3A	:	74	4A	J	90	5A	Z	106	6A	j	122	7A	z
11	0B	VT	27	1B	ESC	43	2B	+	59	3B	;	75	4B	K	91	5B	[	107	6B	k	123	7B	{
12	0C	FF	28	1C	FS	44	2C	,	60	3C	<	76	4C	L	92	5C	\	108	6C	l	124	7C	
13	0D	CR	29	1D	GS	45	2D	-	61	3D	=	77	4D	M	93	5D	]	109	6D	m	125	7D	}
14	0E	SO	30	1E	RS	46	2E	.	62	3E	>	78	4E	N	94	5E	^	110	6E	n	126	7E	~
15	0F	SI	31	1F	US	47	2F	/	63	3F	?	79	4F	O	95	5F	_	111	6F	o	127	7F	DEL

## Codificación de texto

- Decodifique los siguientes mensajes codificados en *UTF-8* y representados en hexadecimal.
  - 41 79 75 64 61
  - 45 6C 20 C3 B1 61 6E 64 C3 BA 20 62 61 6A C3 B3 20 65 6C 20 C3 A1 72 62 6F 6C
  - Para cada uno de los mensajes anteriores, responda: ¿cuántos caracteres posee? ¿cuántos bytes ocupa?
- Codifique su apellido y legajo en *ASCII*, respetando el siguiente formato: “Apellido (legajo)”. Remplace aquellos caracteres que no puedan ser representados por el símbolo “?”.

## Representación de imágenes

Los archivos de imagen utilizados en los ejercicios respetan el siguiente formato:

Ancho	Alto	Bits por pixel	Datos de la imagen
<i>1 byte</i>	<i>1 byte</i>	<i>1 byte</i>	

*Por simplicidad, el formato no incluye la paleta de colores*

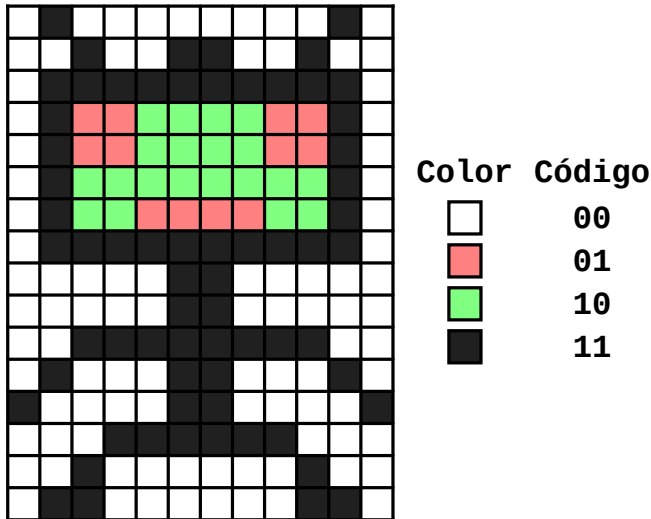
**Ejemplo** dado un archivo de imagen cuyo contenido expresado en hexadecimal es: “**04 06 01 69 12 4F**” y cuyo formato es el descrito en la teoría, para poder obtener la imagen se deben seguir los siguientes pasos:

- Extraer los datos de la cabecera de la imagen: *ancho*, *alto*, y *bits por pixel*:
  - **Ancho:** 4 pixeles.
  - **Alto:** 6 pixeles.
  - **Bits por pixel:** 1 bit por pixel.
- Representar en binario los datos de la imagen:  
0110 1001 0001 0010 0100 1111
- Crear una cuadrícula de *ancho x alto* pixeles.
- Tomando de a “*bits por pixel*” de los datos de la imagen, rellenar la cuadrícula, comenzando desde la esquina superior izquierda, completando las filas:

<i>Datos de la imagen</i>		<i>Imagen</i>
<i>Hex.</i>	<i>Binario</i>	
6	0110	
9	1001	
1	0001	
2	0010	
4	0100	
F	1111	

## Ejercicios

- Sabiendo que el contenido de un archivo de imagen en hexadecimal es: “08 08 01 00 27 65 65 25 25 25 77”, dibuje su imagen.
- Codifique la siguiente imagen expresando el contenido de su archivo en hexadecimal.



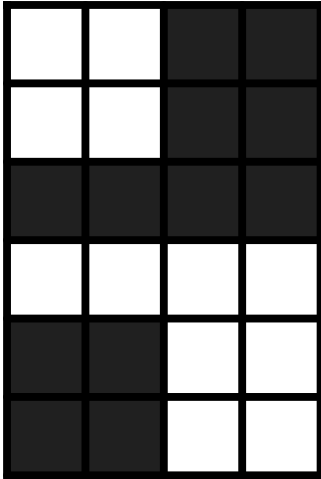
## Compresión

### Compresión con perdida

- Dada la siguiente codificación (representada en **hexadecimal**) que corresponde a una imagen:  
**0C 10 01 40 22 64 7F E7 0E 70 E4 02 4F 27 FE 06 00 60 3F C4 62 86 10 F0  
 10 83 0C**
  - Dibuje la imagen resultante considerando una paleta de 2 colores.
  - ¿Cuántos bytes requiere la codificación dada de la imagen? ¿Y la del ejercicio 4?
  - ¿Que ventajas y desventajas tiene este tipo de codificación?

### Compresión sin perdida

- Considerando la imagen que se muestra abajo, aplique un esquema de compresión que agrupa píxeles consecutivos de igual color y los reemplaza por una codificación “*cantidad/color*”, utilizando una codificación **3+1**, con tres bits para la cantidad y un bit para el color.



*Tenga en cuenta que al calcular la cantidad se debe considerar que las filas de la imagen son consecutivas. Es decir, si una fila termina con dos pixeles negros y la siguiente comienza con otros dos pixeles negros, la codificación debe ser "4 pixeles negros"*

7. Sabiendo que el contenido de un archivo de imagen en hexadecimal es: **"0C 10 02 10 78 07 30 72 0B 20 73 2B 20 72 52 24 72 07 25 22 47 20 78 87 20 72 91 28 72 2B 60 BA 0B 72 33 07 30 B3 07 10 74 0B 40 73 1B 50 76 07 30 B6 0B 10"**, y que la imagen fue codificada utilizando el esquema de compresión "*cantidad/color*", con **4 bits** para representar la cantidad:
- Dibuje la imagen.
  - El formato de imagen presentado en este practico no incluye la paleta de colores ¿Genero esto algún problema al decodificar la imagen?
  - Comparar la cantidad de bytes requeridos para esta codificación frente a las de los ejercicios 4 y 5 ¿Cuál requiere la menor cantidad de bytes?
  - ¿Que ventajas y desventajas tiene este tipo de codificación?