

## Práctica 4 — Representación digital de la información II

Esta práctica abarca los siguientes temas de la Unidad 2:

- a) Representación de textos. Códigos de caracteres: BCD, ASCII, Unicode. Texto con formato: representación sobre texto ASCII o Unicode con *markup* (RTF, HTML,  $\text{\LaTeX}$ ) o binaria. Archivos de texto vs. binarios.
- b) Representación de imágenes, sonido y video. Representación matemática de imágenes. Representación del color. Discretización: pixels y profundidad de color. Compresión con pérdida y sin pérdida. Formatos *bitmap*. Representación vectorial: SVG, Postscript. Representación matemática del sonido. Digitalización del sonido. Video.

**Bibliografía:** Null y Lobur (2003, cap. 2), Wikipedia (2013), González et al. (2009, caps. 2, 6), Li y Drew (2004, cap. 6)

**Problema 1. El código BCD.** Escriba los números 46, 123 y -166 en BCD y BCD empaquetado.

**Problema 2. El código ASCII.**

- a) Escriba su nombre en formato ASCII codificado decimal y hexadecimal.
- b) Utilizando la función `char()`, escriba un programa en Octave/Matlab que muestre todos los caracteres imprimibles del código ASCII junto con su código numérico. ¿Qué sucede si intenta imprimir los códigos del 0 al 31?

Cod	Car
32	
33	!
34	"
35	#
36	\$
37	%
38	&
39	'
40	(
...	

- c) Con un editor de texto cree un archivo de texto ASCII con pocas líneas. Con unas líneas de código como las siguientes, léalo en un vector Octave/Matlab (`v` contiene, después de la operación `fread`, los códigos ASCII de los caracteres leídos). Determine qué convención usa su sistema operativo para indicar el fin de una línea.

```
f=fopen("texto.txt","r");
v=fread(f,Inf);
dec2hex(v)
char(v)
fclose(f);
```

**Problema 3. Unicode.** El archivo `texto-utf8.txt` almacena un texto de pocas líneas en Unicode con codificación UTF-8. Intente abrirlo con un editor de texto para ver su contenido (**Emacs**, por ejemplo, es capaz de interpretarlo correctamente). Luego léalo byte por byte como en el problema anterior e imprima los primeros 20 bytes en binario. Indique a cuántos caracteres corresponden. Para los primeros bytes deberá obtener una salida como la siguiente.

```
11000011
10001001
01110011
01110100
...
```

#### Problema 4. Imágenes.

- Indique cuánta memoria (en bytes) es necesario para almacenar una imagen en los siguientes casos:
  - Imagen monocromática (dos colores) de  $640 \times 480$  pixels.
  - Imagen de 8 colores de  $640 \times 480$  pixels.
  - Imagen de 256 colores de  $1024 \times 768$  pixels.
  - Imagen de  $1024 \times 768$  pixels con profundidad de color de 16 bits.
- Estime el tamaño en MB de una imagen de 8 megapixels almacenada en RGB a 8 bits por canal y compresión al 40%.
- Lea en Octave/Matlab la famosa imagen `Lenna.png` (versión recortada de una foto de la modelo sueca Lena Söderberg). Puede hacerlo con código como el que sigue:

```
img = imread("Lenna.png");
[lx,ly,nc] =size(img);
imshow(img);
```

Este fragmento lee la imagen, la muestra en pantalla y guarda en `lx` y `ly` el tamaño de la misma y en `nc` el número de colores. Puesto que se trata de una imagen de  $512 \times 512$  pixels, con color codificado en RGB, debe obtener `lx=512`, `ly=512`, `nc=3`. A partir de la imagen original, obtenga una imagen sin color rojo, y otra imagen en tonos de grises (transformando apropiadamente la matriz `img`).



**Problema 5.** Calcule la capacidad de almacenamiento (en bytes) necesaria para almacenar 70 minutos de audio en calidad CD-ROM (muestras de 16 bits a 44.1 kHz en dos canales).

**Problema 6.** ¿Cuál es el factor de compresión para un fragmento de audio grabado en calidad CD-ROM y luego comprimido en MP3 a 128 kbps?

**Problema 7.** La cámara de un teléfono celular puede registrar videos con secuencias de imágenes de  $320 \times 240$  a 25 cuadros por segundo y sondeo en MP3 a 128 kbps. Si el factor de compresión de las imágenes es 50%, estimar el tamaño de un fragmento de video de 1 minuto de duración.