

# Ejercicios de procesamiento de vídeo e imagen

## Índice

1 Reproducción de vídeo.....	2
2 Procesamiento de imagen (*).....	2

## 1. Reproducción de vídeo

Vamos en primer lugar a ver cómo crear un reproductor de vídeo en iOS. Tenemos una plantilla en el proyecto `VideoPlayer`. Vemos que tiene un botón *Reproducir vídeo*, y un fichero `video.m4v`. La reproducción de vídeo deberá comenzar en el método `playVideo:`, que es el que se ejecuta al pulsar el botón anterior. Se pide:

- a) Reproducir el vídeo con la pantalla de reproducción predefinida (`MPMoviePlayerViewController`).
- b) Comentar el código implementado en el punto anterior. Ahora vamos a crear un reproductor propio mediante `MPMoviePlayerController`. Crearemos un reproductor de este tipo, haremos que su tamaño sea el mismo tamaño de la vista principal, añadiremos el reproductor a dicha vista como subvista, y comenzaremos la reproducción.
- c) Con el reproductor anterior tenemos el problema de que al finalizar el vídeo el reproductor se queda en pantalla y no hay forma de salir de él. Vamos a escuchar la notificación de reproducción finalizada para que cuando esto ocurra el reproductor sea eliminado de pantalla. Cuando recibamos esta notificación llamaremos al método `videoPlaybackDidFinish:`, que ya se encuentra implementado.
- d) Si giramos el dispositivo veremos que el vídeo no se adapta de forma correcta a la pantalla. Ajustar su propiedad `autoresizingMask` para que sea flexible tanto de ancho como de alto. Comprobar ahora que al girar la pantalla el vídeo se adapta correctamente.
- e) Al reproducir el vídeo en vertical gran parte de la pantalla queda en negro. Vamos a decorar el fondo para darle un mejor aspecto. Crearemos una vista que muestre la imagen `fondo.png`, y la mostraremos como subvista de la vista de fondo del vídeo.
- f) Por último, para que el reproductor quede totalmente integrado en nuestra aplicación, eliminaremos los controles de reproducción que incorpora por defecto. De esta forma el usuario no podrá saltar el vídeo, ni volver atrás en él.

## 2. Procesamiento de imagen (\*)

En este ejercicio procesaremos una imagen con `CoreImage` tanto utilizando la CPU como la GPU. En el proyecto `ProcesamientoImagen` tenemos toda la infraestructura necesaria ya creada. En `viewDidLoad` se inicializa la imagen `CIImage` original, y los contextos CPU y GPU. Tenemos dos *sliders* que nos permitirán aplicar filtros con diferentes niveles de intensidad. En la parte superior de la pantalla tenemos una imagen (`UIImageView`) con un *slider* para aplicar el filtro utilizando la CPU, y en la mitad inferior tenemos una vista `OpenGL` (`GLKView`) y un *slider* para aplicar el filtro en ella utilizando la GPU. Se pide:

- a) Implementar el filtrado utilizando CPU, en el método `sliderCpuCambia:` que se ejecutará cada vez que el *slider* superior cambie de valor. Utilizaremos el filtro de color

sepia (CISepiaTone), al que proporcionaremos como intensidad el valor del *slider*.

b) Implementar el filtrado utilizando GPU, en el método `sliderCpuCambia:` que se ejecutará cada vez que el *slider* inferior cambie de valor. Utilizaremos el mismo filtro que en el caso anterior, pero en este caso guardaremos la imagen resultante en la propiedad `imagenFiltrada` y haremos que se redibuje la vista OpenGL para que muestre dicha imagen. Mueve los dos *sliders*. ¿Cuál de ellos se mueve con mayor fluidez?

c) Vamos a encadenar un segundo filtro, tanto para el contexto CPU como GPU. El filtro será `CIHueAdjust`, que se aplicará justo después del filtro sepia. Consulta la documentación de filtros de Apple para saber qué parámetros son necesarios. Se utilizará el mismo *slider* que ya tenemos para darle valor a este parámetro, es decir, el mismo *slider* dará valor simultáneamente a los parámetros de los dos filtros.

d) Por último, vamos a permitir guardar la foto procesada mediante CPU en el álbum de fotos del dispositivo. Para ello deberemos introducir en el método `agregarFoto:` el código que se encargue de realizar esta tarea, tomando la foto de `self.imageView.image`. Este método se ejecutará al pulsar sobre el botón que hay junto a la imagen superior.

