

Ejercicios de propiedades y colecciones

Índice

1 Propiedades.....	2
2 Listas.....	2
3 Temporizadores (*)......	2
4 Gestión de memoria con ARC.....	3

1. Propiedades

Vamos a añadir propiedades a la clase `UPelicula` creada en la sesión anterior.

- a) Crea propiedades para cada variable de instancia definida en la clase. Las variables de instancia pueden eliminarse, o renombrarse para no confundirse con la propiedad.
- b) En `UAMasterViewController`, haz que la película sea una propiedad de la clase, en lugar de una variable de instancia.
- c) Haz que en cada celda, en lugar de mostrar la descripción de la película muestre sólo el título, accediendo a él a través de la propiedad definida anteriormente.

2. Listas

Vamos ahora a utilizar colecciones de datos en nuestra aplicación. Se pide:

- a) Añade a las películas una lista de actores. Esta lista debe poder ser accedida como propiedad.
- b) Ahora en lugar de guardar una única película vamos a tener una lista de ellas. Para ello en la clase `UAMasterViewController` sustituiremos la propiedad de tipo `UPelicula` por una de tipo `NSArray`, y en el constructor crearemos una serie de películas de prueba y las añadiremos a la lista.
- c) Modifica el método `tableView:numberOfRowsInSection:`, y haz que devuelva el número de componentes de la lista, en lugar de un valor fijo.
- d) Modifica el método `tableView:cellForRowAtIndexPath:`, para que muestre como texto de la celda el título de la película correspondiente a la fila actual. Esta vez, accede a la propiedad `row` de `NSIndexPath` mediante el operador `'.'`. Comprueba que la lista de películas se muestra de forma correcta en la lista.

3. Temporizadores (*)

Crearemos ahora un temporizador con `NSTimer`, que transcurridos unos segundos escribirá un log.

- a) Cuando se lance la aplicación, programa un temporizador que se dispare en 10 segundos. Escribe un log en el momento en el que se programa, y otro en el momento en el que se dispara (dentro del método programado).
- b) Haz ahora que al dispararse el temporizador se envíe una notificación con identificador `"TemporizadorCompletado"`.

- c) En la clase `UAMasterViewController` registra un método como observador de la notificación `"TemporizadorCompletado"`. En dicho método haz que se añada una nueva película a la lista, y actualiza la interfaz llamando a `[self.tableView reloadData]`.
- d) Modifica el observador anterior para utilizar un bloque, en lugar de un método.
- e) Haz que el observador anterior añada a la lista de películas un objeto de tipo `NSString`, en lugar de `UAPelicula`. Al mostrar la lista de películas, cuando se añada la cadena dará un error, ya que estamos intentando mostrar la propiedad `item.titulo`, que no existe en `NSString`. Modificar el código de `tableView:cellForRowAtIndexPath:` para que compruebe si el item de la lista responde al selector `titulo`. De no ser así, mostraremos como texto la descripción del propio objeto.

4. Gestión de memoria con ARC

Vamos a convertir el proyecto anterior a ARC para que la gestión de la memoria la haga automáticamente el compilador.

- a) Realizar un *snapshot* del proyecto antes de realizar la conversión.
- b) Convertir el proyecto de forma automática con la opción *Edit > Refactor > Convert to Objective-C ARC* ¿Qué cambios se producen?

