

# Ejercicios de Xcode y Objective-C

## Índice

1 Creación de un proyecto con Xcode.....	2
2 Manejo de cadenas.....	2
3 Localización.....	2
4 Creación de objetos.....	3
5 Manejo de fechas (*).....	3
6 Gestión de errores (*).....	4

## 1. Creación de un proyecto con Xcode

a) Vamos a crear un nuevo proyecto de tipo *Master-Detail Application*. El nombre de producto será `Filmoteca` y nuestra organización `es.ua.jtech`. Utilizaremos como prefijo para todas las clases `UA`, y sólo destinaremos la aplicación al iPhone por el momento. También desmarcaremos todas las casillas adicionales, ya que no vamos a utilizar por el momento ninguna de esas características (Core Data, ARC, pruebas de unidad, storyboards).

b) ¿Qué versión de iOS requiere la aplicación? Modifica la configuración necesaria para que funcione en dispositivos con iOS 4.0.

c) Ejecuta la aplicación en el simulador de iPhone. Comprueba que funciona correctamente.

## 2. Manejo de cadenas

Vamos a cambiar las cadenas de la aplicación. Se pide:

a) En la clase `UAMasterViewController` localiza el método `initWithNibName:bundle:`. Hay una línea en la que se asigna el título mediante la instrucción `self.title = ....` Modifica el título para que ahora sea "Filmoteca".

b) Abre la documentación de la clase `NSString`. Busca un método que permita convertir la cadena a mayúsculas, y aplícalo al título.

c) Vamos a hacer que en la lista aparezcan varios items. Para ello localiza el método `tableView:numberOfRowsInSection:`, y haz que devuelva el valor 5. Comprueba que ahora aparecen cinco items en la lista.

d) Ahora cambiaremos el texto de cada item, para que indique la posición de la lista en la que está. Por ejemplo, el primer item tendrá el texto `Posición 0`, el segundo `Posición 1`, etc. Para ello localiza el método `tableView:cellForRowAtIndexPath:`, y dentro de él la línea que comienza por `cell.textLabel.text = ....` Aquí es donde se asigna el texto de cada item. La posición del item que se está asignando se puede obtener llamando al método `row` del objeto `NSIndexPath`, y es un valor de tipo entero (consulta la documentación de la clase `NSIndexPath` en Organizer).

## 3. Localización

Vamos ahora a localizar los textos de nuestra aplicación. Se pide:

a) Añade a la aplicación las localizaciones a inglés y español.

- b) Haz que el nombre que se muestra bajo el icono de la aplicación en el iPhone cambie según el idioma del dispositivo (`Filmoteca`, `FilmLibrary`)
- c) No queremos localizar los ficheros `.xib`. Elimina la localización de dichos ficheros.
- d) Crea un fichero `Localizable.strings`. Localiza a español e inglés las cadenas introducidas en el ejercicio anterior utilizando este fichero.

## 4. Creación de objetos

Vamos a crear una clase para representar las películas. La clase tendrá como nombre `UAPelicula`, y contendrá los siguientes datos:

- Título
- Director
- Calificación de edad (puede ser TP, NR7, NR13 o NR18)
- Puntuación (número real de 0 a 10)

- a) Crea la clase, e introduce las variables de instancia necesarias para representar los datos anteriores, utilizando el tipo de datos que consideres más adecuado.
- b) Implementa un inicializador sin parámetros, otro que lleve sólo el título, y otro que incluya todos los datos. ¿Cuál será el inicializador designado? Impleméntalo teniendo esto en cuenta.
- c) Crea un método factoría para cada inicializador.
- d) Sobrescribe el método `description` para que imprima el título de la película, seguido del nombre del director entre paréntesis. Por ejemplo: `El Resplandor (Stanley Kubrick)`.
- e) Modifica la clase `UAMasterViewController` para que en su inicializador se instancie un objeto de tipo película, y en su `dealloc` se libere su memoria. Como texto de las celdas, mostraremos la descripción del objeto `UAPelicula` creado.

## 5. Manejo de fechas (\*)

Vamos a añadir a la clase `UAPelicula` la fecha de estreno de la película. Se pide:

- a) Añadir una nueva variable de instancia para la fecha de estreno de la película. Modificaremos los inicializadores para incluir este dato adicional en el inicializador designado.
- b) Modificamos el método `description` para que la película ahora muestre junto al nombre del director, el año de su estreno. Por ejemplo, `El Resplandor (1980, Stanley Kubrick)`. En el campo fecha realmente tendremos almacenada la fecha exacta

(día, mes y año).

c) Añadir un método que nos devuelva la antigüedad de la película en años. Por ejemplo, si la película se estrenó en 1980 y estamos en 2011, nos devolverá 31 años. Escribir un *log* con la antigüedad tras instanciar la clase en el inicializador de `UAMasterViewController`.

## 6. Gestión de errores (\*)

Vamos a ver como tratar errores al llamar a métodos que pueden fallar por factores externos. Utilizaremos el método `stringWithContentOfFile:encoding:error:` de la clase `NSString` para este ejercicio. Debemos consultar la documentación de este método en Organizer. Se pide:

a) Crear dos ficheros de texto: `texto1.txt` y `texto2.txt` que contengan el texto `Primer fichero` y `Segundo fichero` respectivamente. El primero de ellos lo empaquetaremos en el raíz del *bundle*, y el segundo en una carpeta `/datos`.

### Ayuda

Para obtener la ruta de un fichero en el sistema de ficheros del dispositivo, puedes utilizar el método `[[NSBundle mainBundle] pathForResource:@"nombreFichero" ofType:@"extension"]`.

b) En el inicializador de `UAMasterViewController` crearemos una cadena a partir de estos ficheros, y escribiremos un *log* con el contenido leído.

c) Intenta leer el segundo fichero con la ruta `/texto2.txt`. Deberá dar error, ya que este fichero se empaqueta en `/datos/texto2.txt`. Utiliza el objeto `NSError` para detectar el error y escribe un *log* con el motivo del error.

