

# Guías de estilo y personalizaciones avanzadas - Ejercicios

## Índice

1 Personalizando las celdas de un Table View (1).....	2
2 Personalizando las celdas de un Table View (2).....	3
3 Personalizando un Tool Bar.....	3

## 1. Personalizando las celdas de un Table View (1)

a) Creamos un nuevo proyecto en XCode que llamaremos CeldasPersonalizadas el cual mostrará un listado de películas. El proyecto va a ser de tipo *Single View Application*, organización es.ua.jtech y prefijo UA. Seleccionamos iPhone como "target" y 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) Abrimos la vista UAViewController.xib y añadimos una vista de tabla Table View a la vista principal. La ajustamos para que ocupe toda la pantalla.

c) Creamos el Outlet necesario dentro de la clase UAViewController y lo relacionamos dentro de la vista. Modificamos la declaración de la clase para añadirle los protocolos delegados de UITableViewController. Relacionamos los delegados de la vista de la tabla con la clase.

d) Creamos la clase para la celda personalizada. A la clase la llamaremos CeldaView, será subclase de UITableViewCell y modificaremos su vista para añadirle los estilos que deseemos (labels, imágenes, botones, etc, etc). La celda debe de tener al menos dos labels y una imagen.

e) Modificamos la controladora de la vista de la celda (CeldaView) añadiendo los Outlets necesarios para las labels, imágenes, etc que hayamos añadido a la vista de la celda y los relacionamos en la vista.

f) Definimos los métodos delegados necesarios para gestionar la tabla, estos métodos serán los siguientes:

```
- (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:
(NSIndexPath *)indexPath {
    //devolvemos altura de la celda
}

// Customize the number of sections in the table view.
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    //devolvemos el numero de secciones (1)
}

- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section
{
    //devolvemos el numero de filas por seccion (10)
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:
(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"TableViewCell";
```

```
CeldaView *cell = (CeldaView*)[tableView
dequeueReusableCellWithIdentifier:
CellIdentifier];
if (cell == nil) {
    //Completamos con el código necesario para cargar la vista
de la celda
}
// Configuramos la celda
return cell;
}
```

g) Arrancamos la aplicación y comprobamos que todo funciona correctamente.

## 2. Personalizando las celdas de un Table View (2)

a) Seguimos con el ejercicio anterior. Vamos a "llenar" de datos las celdas de la tabla que hemos creado, para ello vamos a crear una clase a la que llamaremos *Pelicula* a la cual contendrá un *NSString* que será el título, otro que será la sinopsis y una imagen (*UIImage*) que será la carátula.

b) Ahora crearemos un objeto de tipo *NSMutableArray* dentro de la clase *UITableViewController*, lo inicializaremos y lo completaremos con al menos 5 películas que queramos. Cada elemento del array será de tipo *Pelicula*.

c) Una vez creado el array de películas vamos a mostrarlas en nuestra tabla, para ello deberemos de completar los métodos de la clase *UITableView Delegate* y completar los datos de las celdas correctamente.

d) Cuando hayamos terminado, comprobamos que la aplicación funciona según lo esperado.

## 3. Personalizando un Tool Bar

En este ejercicio deberemos de personalizar a nuestro gusto un elemento de tipo *UIToolBar*. Comenzamos creando un proyecto nuevo en XCode que llamaremos *ToolBarPersonalizada*. El proyecto será de tipo *Single View Application* para iPhone. Cuando tengamos el proyecto creado abrimos la vista principal y arrastramos un *Tab Bar View* desde el navegador del Interface Builder.

De forma programada tendremos que personalizar el *Tab Bar* añadiéndole al menos un botón, un *Segmented Control* y color o imagen de fondo.

