

# Tema 10

## Creación de Interfaces Gráficas Parte III

# UITableView

- Clase pensada para mostrar información en formato de lista
  - Subclase de UIScrollView
  - Sólo puede mostrar una columna
  - Personalización a través de su *dataSource* y su *delegate*
  - Dos estilos posibles (Plain y Grouped)

# UITableView

- ¿Cómo crear una UITableView?
  - Utilizando el Interface Builder
  - `alloc/initWithFrame:style:`
- Una vez creada hay que asociarle un *dataSource* para proporcionarle a la tabla el contenido de las celdas.

# UITableViewDataSource

- Número de secciones en la tabla

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView;
```

- Número de filas en una sección

```
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionSection:  
    (NSInteger)section;
```

- Contenido de una celda concreta

```
- (UITableViewCell *)tableView:(UITableView *)tableView  
    cellForRowAtIndexPath:(NSIndexPath *)indexPath;
```

# UITableViewCell

- Clase que representa el contenido de cada una de las celdas en una UITableView
- Propiedades importantes:

```
@property(n nonatomic, readonly, retain) UILabel *textLabel;  
@property(n nonatomic, readonly, retain) UILabel *detailTextLabel;  
@property(n nonatomic, readonly, retain) UIImageView *imageView;  
@property(n nonatomic) UITableViewCellAccessoryType accessoryType;
```

# UITableViewDelegate

- Se encarga de controlar el **comportamiento** de la UITableView, no su contenido (el contenido lo controla el dataSource)

- Método más importante:

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath;
```

# UITableViewController

- Subclase de `UIViewController`.
- Preparada para funcionar como *dataSource* y *delegate* de una `UITableView`.
- Implementa el método *loadView*, y se encarga de instanciar una `UITableView` de manera automática.
- Automatiza otros comportamientos típicos del *delegate*.
- Normalmente instanciaremos esta clase **sin** un `.xib` de apoyo.

# UIPageViewController

- Subclase de **UIViewController**.
- Permite a los usuarios navegar entre view controllers usando una transición específica.
- La navegación es controlada por los usuarios utilizando gestos de desplazamiento de pantalla.



# UIPageViewController

- Se puede definir las esquinas (**spines**) desde las cuales se puede cambiar de página.

## – Horizontalmente:

- `UIPageViewControllerSpineLocationNone`
- `UIPageViewControllerSpineLocationMin`: (Esquina Izquierda)
- `UIPageViewControllerSpineLocationMid`: (Medio)
- `UIPageViewControllerSpineLocationMax`: (Esquina derecha)

## – Verticalmente:

- `UIPageViewControllerSpineLocationNone`
- `UIPageViewControllerSpineLocationMin`: (Arriba)
- `UIPageViewControllerSpineLocationMid`: (Medio)
- `UIPageViewControllerSpineLocationMax`: (Abajo)

# UIAlertView

- Esta clase es la encargada de mostrar un mensaje de alerta al usuario.
- Se puede definir el título de la alerta, así como su mensaje. También puedes añadir botones.

# UIAlertView

- Existen diferentes estilos de alerta:
  - **UIAlertViewDefault**
    - Alerta estándar.
  - **UIAlertViewSecureTextInput**
    - Alerta que permite la entrada de texto con el campo de texto oscurecido.
  - **UIAlertViewPlainTextInput**
    - Alerta que permite la entrada de texto.
  - **UIAlertViewLoginAndPasswordInput**
    - Alerta con identificador y password.

# Customizando controles de UIKit

- En iOS 6 se puede customizar fácilmente la apariencia de varios controles de UIKit.
- Podemos cambiar el color de los elementos, la imagen de fondo, la posición de los títulos en muchos objetos, incluyendo *toolbars*, *navigation bars*, *search bars*, *buttons*, *sliders* y otros controles.

# Customizando controles de UIKit

- **setBackgroundImage:** Cambia la imagen de fondo de un objeto.
- **setTitlePositionAdjustment:** Ajusta la posición del título.
- **setTintColor:** Cambia la apariencia del objeto “switch” dependiendo de la posición en la que esté.