



Curso de especialización

Programación de
Aplicaciones para iPhone/
iPad de Apple

Octubre-Diciembre 2011

Ángel Díaz

angel@blueplastic.net

Contenidos Lección 4

- Objective-C
 - *NSArray y NSMutableArray*
- Cocoa Touch
 - *Animaciones en el cambio de vista*
 - *UIAlertView*
 - *UIAlertViewDelegate*
 - *UITableView*
 - *UITableViewDelegate*
 - *UITableViewDataSource*
- Patrones de Diseño
 - *ModalViewController*
- Xcode
 - *Aplicaciones MultiView*
 - *Aplicaciones Multi-Idioma*





Curso de especialización

Programación de
Aplicaciones para iPhone/
iPad de Apple

Patrones de Diseño: *ModalViewController*

Ángel Díaz

angel@blueplastic.net

ModalViewController (I)

- Se utiliza para la presentación de vistas en la interfaz de usuario de las aplicaciones.
- Pila (*Stack*)
- Ejemplo de uso:
 - Pantalla de menú

ModalViewController (II)

- Mostrar un *ViewController* (apilar - PUSH):
 - `[self presentViewController:one animated:YES];`
- Quitar un *ViewController* (quitar de la pila - POP):
 - `[self dismissViewControllerAnimated:YES];`

ModalViewController (III)

menuViewController

Pantalla N

...

Pantalla 2

Pantalla 1



Curso de especialización

Programación de
Aplicaciones para iPhone/
iPad de Apple

Xcode: Aplicación *MultiView*

Ángel Díaz

angel@blueplastic.net

Aplicación *MultiView* (I)

- Obviamente nuestras aplicaciones pueden tener diferentes vistas/interficies.
- Requiere pensar un poco sobre el orden que van a tener las diferentes vistas en la aplicación y como se va a efectuar la interacción.
- Varias opciones:
 - 1 ViewController – N Views
 - 1 ViewController – 1 View

Aplicación *MultiView* (II)

- Diferentes formas de mostrar Vistas:
 - Añadir como *subView* una *view*.
 - `[self.view addSubview:view1];`
 - Intercambiar la *view* principal.
 - `self.view = view1;`
 - *ModalViewController*
 - `[self.view
presentModalViewController:view1
animated:YES];`

Aplicación *MultiView* (III)

- 1 ViewController – N Views
- Opción recomendable para diferentes tipos de visualizaciones de un mismo contenido.
- Abrir
 - `[self.view addSubview:view1];`
- Cerrar
 - `[self.view removeFromSuperview];`

Ejemplo X

- Aplicación sencilla.
 - Label
 - Imagen
 - Botón
 - 2 Views
- View 1 muestra la label y la imagen dando prioridad a la imagen sobre la label.
- View 2 muestra la label y la imagen dando prioridad a la label sobre la imagen.
- Botón Cambia entre las Views.

Aplicación *MultiView* (IV)

- 1 ViewController – 1 Views
- Recomendable para diferentes tipos de contenidos.
- Abrir
 - `[self presentViewController:view animated:YES];`
- Cerrar
 - `[self dismissViewControllerAnimated:YES];`

Ejemplo XI

- Aplicación sencilla.
 - Label
 - Botón
 - 2 *ViewControllers*
 - Imagen
 - *TextView*
- *ViewController 1* muestra la label y la imagen.
- *ViewController 2* muestra la label y el *TextView*.

Ejercicio VI

- Aplicación sencilla.
 - View 1:
 - Inputs: Nombre, Apellidos, Número de Teléfono
 - Botón Previsualizar
 - View 2:
 - Label: Nombre, Apellidos, Número de Teléfono
 - Boton Guardar
- Guardar Nombre, Apellidos, Número de Teléfono sin persistencia.
- Boton Previsualizar abre View 2 cargando los datos de la persistencia.



Curso de especialización

Programación de
Aplicaciones para iPhone/
iPad de Apple

Cocoa Touch : Animaciones en el cambio de vista

Ángel Díaz

angel@blueplastic.net

Animaciones en el cambio de vista (I)

- addSubview
 - UIViewAnimationTransitionFlipFromLeft
 - UIViewAnimationTransitionFlipFromRight
 - UIViewAnimationTransitionCurlUp
 - UIViewAnimationTransitionCurlDown



Animaciones en el cambio de vista (II)



- addSubview
 - ```
[UIView beginAnimations:nil context:NULL];
[UIView setAnimationDuration:2.0];
[UIView
setAnimationTransition:UIViewAnimationTransitionFl
ipFromLeft forView:tmp cache:YES];
[self.view addSubview:newview];
[UIView commitAnimations];
```

## *Ejemplo XII*

- Aplicación sencilla.
  - View 1
    - Label 1 y Botón 1
  - View 2
    - Label 2 y Botón 2
- Botón 1 abre View 2 y botón 2 abre View 1.
- **Animando con transición el cambio de vista.**

# *Animaciones en el cambio de vista (III)*

- presentModalViewController
  - UIModalTransitionStyleCoverVertical
  - UIModalTransitionStyleFlipHorizontal
  - UIModalTransitionStyleCrossDissolve
  - UIModalTransitionStylePartialCurl



# Animaciones en el cambio de vista (IV)



- presentModalViewController

```
newViewController *newView = [[newViewController alloc]
 initWithNibName:@"newViewController" bundle:nil];
newView.modalTransitionStyle =
 UIModalTransitionStyleCrossDissolve;
[self presentModalViewController:newView animated:YES];
```

## *Ejemplo XIII*

- Aplicación sencilla.
  - *ViewController 1*:
    - Label 1 y Botón 1
  - *ViewController 2*:
    - Label 2 y Botón 2
- Botón 1 abre View 2 y botón 2 abre View 1.
- **Animando con transición el cambio de vista.**

## Ejercicio VII

- Aplicación sencilla.
  - View 1:
    - Label, Botón 1, Botón 2
  - View 2:
    - Imagen, Botón 3
  - View 3:
    - Texto, Botón 4
- Botón 1 abre View 2.
- Botón 2 abre View 3.
- Botón 3 y Botón 4 vuelven a View 1.

Curso de especialización

Programación de  
Aplicaciones para iPhone/  
iPad de Apple

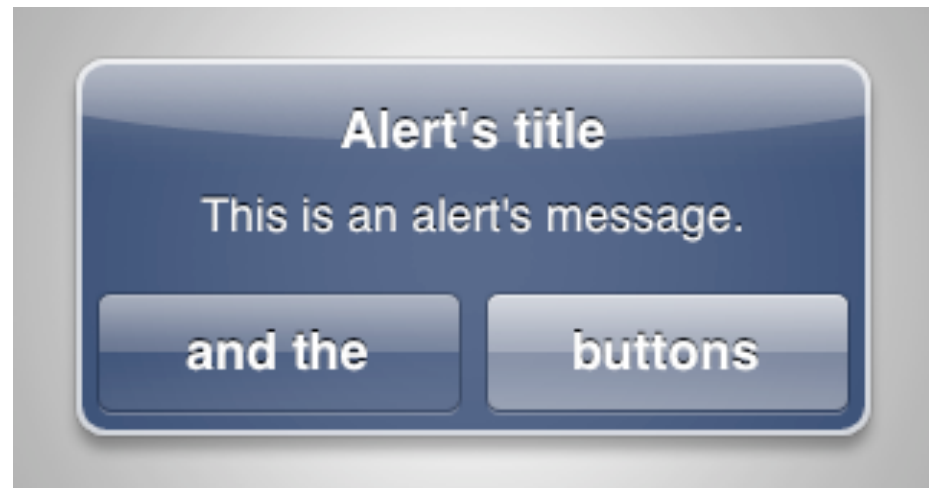
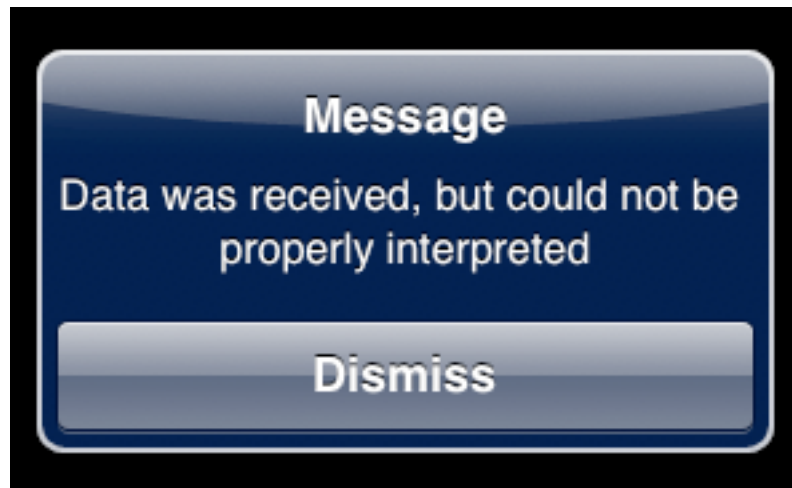
## Cocoa Touch : *UIAlertView*

Ángel Díaz

[angel@blueplastic.net](mailto:angel@blueplastic.net)

# *UIAlertView (I)*

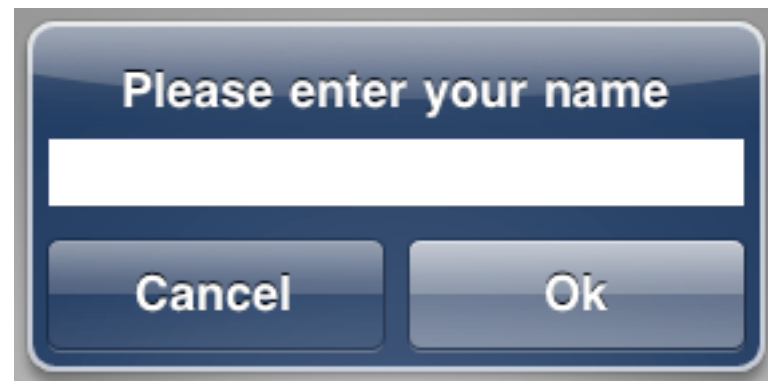
- UIAlertViewDelegate





## *UIAlertView (II)*

- Requieren la atención / interacción del usuario
- Usos:
  - Informar al usuario
  - Plantear pregunta/decisión
  - Pedir información



## *UIAlertView (III)*

- *Alert Simple*

```
UIAlertView *alert = [[UIAlertView alloc]
 initWithTitle:nil
 message:@"An Alert!"
 delegate:nil
 cancelButtonTitle:@"OK"
 otherButtonTitles:nil];
[alert show];
[alert release];
```

## *UIAlertView (IV)*

- (ó bien) [ iOS < 5]

```
[[[[UIAlertView alloc] initWithTitle:@"Alert
Title"
message:@"Alert Message"
delegate:nil
cancelButtonTitle:@"Cancel"
otherButtonTitles:nil] autorelease] show];
```

## *UIAlertView (V)*

- (ó bien) [ iOS 5]

```
[[[UIAlertView alloc] initWithTitle:@"Alert
Title"
message:@"Alert Message"
delegate:nil
cancelButtonTitle:@"Cancel"
otherButtonTitles:nil] show];
```

## *UIAlertView (VI)*

- *Alert con Varios Botones*

```
UIAlertView *alert = [[UIAlertView alloc]
 initWithTitle:nil
 message:@"An Alert!"
 delegate:nil
 cancelButtonTitle:@"OK"
 otherButtonTitles:@"Cancel",nil];
[alert show];
[alert release];
```

## *UIAlertViewDelegate (I)*

- Botón Seleccionado
  - `alertView:clickedButtonAtIndex:`
- Cancelado
  - `alertViewCancel`
  - Se usa poco ya que en `alertView:clickedButtonAtIndex` también puedes obtener cuando se presiona el botón de cancel.

## *UIAlertViewDelegate (II)*

```
- (void) alertView:(UIAlertView *)alertView
 clickedButtonAtIndex:(NSInteger)buttonIndex {
 if (buttonIndex == 0) {
 //cancel

 //do something
 } else if (buttonIndex == 1) {
 //other button

 //do anything
 }
}
```

## *Ejemplo XIV*

- Aplicación sencilla.
  - *View*
    - *Label*
    - Botón 1
  - *Alert*
    - Botón 2
    - Botón 3
- Botón 1 abre *Alert* con 2 botones
- Botón 2 escribe algo en *Label*
- Boton 3 limpia *Label*



## Ejercicio VIII

- Aplicación sencilla.
  - View 1:
    - Label, Botón 1
  - Alert 1:
    - Botón 2, Botón 3
  - View 2:
    - Imagen, Botón 4
- Botón 1 abre *Alert* 1.
- Botón 2 abre *View* 2.
- Botón 3 no hace nada (*Cancel*).
- Botón 4 vuelve a *View* 1.



Curso de especialización

Programación de  
Aplicaciones para iPhone/  
iPad de Apple

## Xcode: Aplicación Multi-Idioma

Ángel Díaz

[angel@blueplastic.net](mailto:angel@blueplastic.net)

# Aplicación Multi-Idioma (I)

- Dos métodos:
  - Ficheros **strings**
    - Fichero de texto clave – valor
    - 1 sólo fichero de *strings* por idioma
  - Diferentes interfaces por idioma (**.xib**)
    - Multiplicar cada interfaz por el número de total idiomas

# Aplicación Multi-Idioma (II)

- Ficheros **strings**
  - Ficheros de texto.
  - Método por defecto para trabajar con traductores.
  - Permite control de versiones, y comparación entre versiones.
  - Puede acarrear algún problema con las interfaces en los diferentes idiomas (textos más largos).
  - Requiere buen testeo.
  - Centralizado.

# Aplicación Multi-Idioma (II)

- Ficheros **strings**
  - Ficheros de texto.
  - Método por defecto para trabajar con traductores.
  - Permite control de versiones, y comparación entre versiones.
  - Puede acarrear algún problema con las interfaces en los diferentes idiomas (textos más largos).
  - Requiere buen testeo.
  - Centralizado.
- **Mantenible**

## *Aplicación Multi-Idioma (III)*

- Diferentes interfaces por idioma (**.xib**)
  - Ficheros “binarios”.
  - No salva 100% el problema de la “localización”, pueden quedar contenidos pendientes de localizar.
  - Disperso.
  - Poco trivial al comparar diferentes versiones.

## *Aplicación Multi-Idioma (III)*

- Diferentes interfaces por idioma (**.xib**)
  - Ficheros “binarios”.
  - No salva 100% el problema de la “localización”, pueden quedar contenidos pendientes de localizar.
  - Disperso.
  - Poco trivial al comparar diferentes versiones.
  
- **Poco mantenible**

## *Aplicación Multi-Idioma (IV)*

- Cómo “localizamos” un fichero?
  - New -> Resources -> String File
    - Nombre **.strings**
  - Seleccionar Fichero a “localizar”
  - Pestaña (Izquierda) File Inspector
  - Apartado Localization
    - Añadir lenguajes (+)



# *Aplicación Multi-Idioma (V)*

- Cómo “localizamos” un contenido?
  - NSLocalizedString
    - **Se basa en el lenguaje que tiene el dispositivo en Ajustes->General->Localización**

- Ejemplo:

```
NSString *myString =
 NSLocalizedString(@"First Name", @"Used
 to ask the user his/her first name");
```

# Aplicación Multi-Idioma (VI)

- Qué aspecto tiene un fichero **.strings**?

...

```
/* comentario que ayuda a comprender el texto que se ha de traducir */
```

```
“llave”=“valor de la llave en el idioma especificado”;
```

```
/* otro comentario que también ayuda */
```

```
“otra llave”=“otro valor”;
```

...

## *Ejemplo XV*

- Aplicación *Multi-View*
  - View 1
    - 1 Label
- Cargar un contenido en *Label* en función del idioma del dispositivo.
- Cambiar el idioma del dispositivo para ver como cambia el texto de la *Label*.

# Aplicación Multi-Idioma (VII)

## ■ Extraball

- Usar NSLocalizedString tiene premio.
- El método de **strings**, recibe el soporte de la herramienta **genstrings**
- **genstrings?**

# Aplicación Multi-Idioma (VIII)



## ■ Extraball

### ■ genstrings

- Herramienta para *Terminal* que recorre los archivos que le especifiquemos en su llamada en busca de :
  - NSLocalizedString
- Genera como resultado un fichero .strings con todas las parejas llave-valor que ha encontrado en nuestro código y además reconoce los comentarios.

## *Ejemplo XVI*

- Ejemplo XV
  - Apartaremos del proyecto los ficheros de **.strings** localizados que hemos generado
  - Llamaremos a genstrings
  - Compararemos el resultado de llamar a genstrings con el que nosotros habíamos compuesto.

Curso de especialización

Programación de  
Aplicaciones para iPhone/  
iPad de Apple

## **Objective-C: *NSArray* & *NSMutableArray***

Ángel Díaz

[angel@blueplastic.net](mailto:angel@blueplastic.net)

# Arrays Objective-C (I)

- *NSArray*
- *NSMutableArray*



## Arrays Objective-C (II)

- *NSArray*
  - Array que no puede ni debe ser modificado una vez inicializado
- *NSMutableArray*
  - Array que puede ser modificado de forma posterior a su inicialización.

## Arrays Objective-C (II)

- *NSArray*

```
NSArray *arr = [NSArray arrayWithArray:
 (NSArray *)];
```

```
NSArray *arr = [NSArray arrayWithObjects:
 (id), ..., nil];
```

...

- *NSMutableArray*

```
[arr addObject:(id)]
```

```
[arr addObjectsFromArray:(NSArray *)];
```

...

## *Arrays Objective-C (II)*

- *Recoger valor*  
`[arr objectAtIndex:(NSUInteger)];`
- *Número de elementos*  
`[arr count];`



## *Alguna cosa més (I)*

- *NSDictionary*
  
- *NSMutableDictionary*

## *Alguna cosa més (II)*

- *NSDictionary*

```
NSDictionary *dict = [NSDictionary
 dictionaryWithDictionary:(NSDictionary *)];
```

```
NSDictionary *dict = [NSDictionary
 dictionaryWithObject:(id) forKey:(id)];
```

```
NSDictionary *dict = [NSDictionary
 dictionaryWithObjectsAndKeys:(id), ..., nil];
```

...

- *NSMutableDictionary*

```
[dict setValue:(id) forKey:(NSString *)];
```

...

## *Alguna cosa más (III)*

- *Recoger valor*  
`[dict valueForKey:(NSString *)];`
- *Número de elementos*  
`[dict count];`



Curso de especialización

Programación de  
Aplicaciones para iPhone/  
iPad de Apple

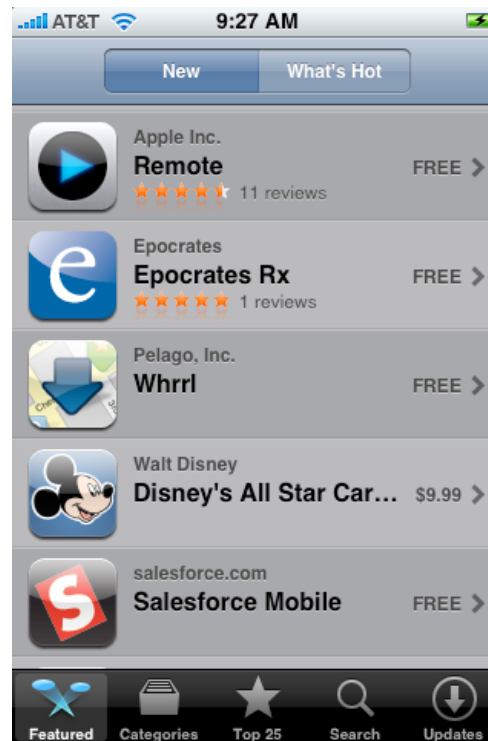
## Cocoa Touch : *UITableView*

Ángel Díaz

[angel@blueplastic.net](mailto:angel@blueplastic.net)

# UITableView (I)

- **UITableViewDelegate**
- **UITableViewDataSource**





# UITableView (II)



## *UITableView (III)*

- *numberOfRowsInSection*
- *numberOfSections*
- *cellForRowAtIndexPath*
- *reloadData*
- ...

# *UITableView (IV)*

- ***UITableViewDelegate***
  - tableView:viewForHeaderInSection:
  - tableView:viewForFooterInSection:
  - tableView:heightForHeaderInSection:
  - tableView:heightForFooterInSection:
  - ...

# *UITableView (V)*

- ***UITableViewDataSource***
  - tableView:cellForRowAtIndexPath:
  - numberOfSectionsInTableView:
  - tableView:numberOfRowsInSection:
  - tableView:titleForHeaderInSection:
  - tableView:titleForFooterInSection:
  - ...

# *UITableView (VI)*

- *Continuará*

