idEC
UNIVERSITAT POMPEU FABRA

Curso de especialización

Programación de Aplicaciones para iPhone/ iPad de Apple

Octubre-Diciembre 2011

Ángel Díaz

angel@blueplastic.net

Contenidos Lección 3

- Cocoa Touch
  - *UITextField* (continuación Lección 2)
  - *UITextFieldDelegate*
- Objective-C
  - Manejo de Memoria
- Patrones de Diseño
  - *Singleton*
- Cocoa
  - *User Defaults*

idEC
UNIVERSITAT POMPEU FABRA

Curso de especialización

Programación de Aplicaciones para iPhone/ iPad de Apple

Cocoa Touch : *UITextField (cont.)*

Ángel Díaz

angel@blueplastic.net

idEC UNIVERSITAT POMPEU FABRA

Lección 3

Ejemplo VI

- Aplicación sencilla:
  - Input
  - Label

- Mientras escribimos en el input se irá actualizando el contenido del label.
- Cuando acabemos de editar el input debe desaparecer el teclado.

idEC UNIVERSITAT POMPEU FABRA

Lección 3

UITextField (IV)

- Seguimos teniendo un problema con el UITextField.
  - *Al editar el UITextField el teclado nos tapa parte de la pantalla.*

- Claves:
  - Escuchar eventos/notificaciones teclado
  - Mover la vista

UITextField (V)

```
- (void)registerForKeyboardNotifications  {

  [[NSNotificationCenter defaultCenter] addObserver:self
       selector:@selector(keyboardWasShown:)
       name:UIKeyboardDidShowNotification
       object:nil];

  [[NSNotificationCenter defaultCenter] addObserver:self
       selector:@selector(keyboardWasHidden:)
       name:UIKeyboardDidHideNotification
       object:nil];

}
```

UITextField (VI)

Lección 3

```
- (void)keyboardWasShown:(NSNotification *)aNotification {
   if ( self.keyboardShown )
          return;
   NSDictionary *info = [aNotification userInfo];
   NSValue *aValue = [info objectForKey:UIKeyboardFrameEndUserInfoKey];
   CGSize keyboardSize =
   [aValue CGRectValue].size;
   NSTimeInterval animationDuration = self.duration;
   CGRect frame = self.view.frame;
```

UITextField (VI cont.)


```
    frame.origin.y -= keyboardSize.height-self.offsetoriginy;
    frame.size.height += keyboardSize.height-self.offsetframeheight;
    [UIView beginAnimations:@"ResizeForKeyboard" context:nil];
    [UIView setAnimationDuration:animationDuration];
    self.view.frame = frame;
    [UIView commitAnimations];
    self.viewMoved = YES;
    self.keyboardShown = YES;
}
```

UITextField (VII)

Lección 3

```
- (void)keyboardWasHidden:(NSNotification *)aNotification {
  if ( self.viewMoved ) {
        NSDictionary *info = [aNotification userInfo];
        NSValue *aValue = [info objectForKey:UIKeyboardFrameEndUserInfoKey];
        CGSize keyboardSize = [aValue CGRectValue].size;
        NSTimeInterval animationDuration = self.duration;
        CGRect frame = self.view.frame;
        frame.origin.y += keyboardSize.height-self.offsetoriginy;
        frame.size.height -= keyboardSize.height-self.offsetframeheight;
```

UITextField (VII cont.)


```
            [UIView beginAnimations:@"ResizeForKeyboard" context:nil];
            [UIView setAnimationDuration:animationDuration];
            self.view.frame = frame;
            [UIView commitAnimations];
            self.viewMoved = NO;
    }
    self.keyboardShown = NO;
}
```

UITextField (VIII)

- Valores para inicialización:
  - Duration = 0.300000011920929
  - Offsetframeheight = 44
  - Offsetoriginy = 144

idEC UNIVERSITAT POMPEU FABRA

Lección 3

Ejemplo VII

- Ejemplo V
- Centrar la vista al introducir datos en el Input y devolverla a su estado anterior al acabar de editar el Input.

idEC UNIVERSITAT POMPEU FABRA

Lección 3

Ejercicio II

- Aplicación sencilla:
  - 3 Inputs
  - 1 Botón

- Al acabar de editar cada uno de los campos el teclado debe desaparecer.
- Se debe recolocar la vista cuando se edite cada uno de los campos.
- Botón limpia el contenido de los 3 Inputs.

Curso de especialización

Programación de Aplicaciones para iPhone/ iPad de Apple

Objective-C : *Manejo de Memoria*

Ángel Díaz

angel@blueplastic.net

idEC UNIVERSITAT POMPEU FABRA

Lección 3

Manejo de Memoria (I)

- ¿ Qué hemos de liberar ?
  - Alloc
  - Copy
  - New
  - @property (nonatomic, retain) …

- ¿ Dónde ?
  - Una vez acabemos de usarlo.
  - dealloc

Manejo de Memoria (II)

Lección 3

```
…
NSString *newString = [[NSString
  alloc] initWithFormat:@”This is
  a string”];
…
statusText.text = newString;
…
```

Manejo de Memoria (II)


```
…
NSString *newString = [[NSString
  alloc] initWithFormat:@”This is
  a string”];
…
statusText.text = newString;
…
[newString release];
```

Manejo de Memoria (III)

idEC UNIVERSITAT POMPEU FABRA

Lección 3

```
…
NSString *newString =
  [stringWithFormat:@”This is a
  string”];
…
statusText.text = newString;
…
```

Manejo de Memoria (III)

idEC UNIVERSITAT POMPEU FABRA

Lección 3

```
…
NSString *newString =
  [stringWithFormat:@”This is a
  string”];
…
statusText.text = newString;
…

//no hay que liberar nada!
```

Manejo de Memoria (IV)

- .h:

```
…
@property (nonatomic, retain) UITextField
 *campo;
…
```

Manejo de Memoria (IV)

idEC UNIVERSITAT POMPEU FABRA

Lección 3

- .m:

```
- (void) dealloc {
  [campo release];
}
…
- (void) viewDidUnload {
  self.campo = nil;
  [super viewDidUnload];
}
```

Curso de especialización

Programación de Aplicaciones para iPhone/ iPad de Apple

Patrones de Diseño: Singleton

Ángel Díaz

angel@blueplastic.net

idEC UNIVERSITAT POMPEU FABRA

Lección 3

Singleton

- Instancia única y compartida
- Con normalidad se utiliza:
  - Para evitar instanciar un mismo objeto desde diferentes puntos del código.
  - Tener un objeto siempre vivo y configurado en memoria.
  - Acceder de forma única y/o sincronizada a datos.

idEC UNIVERSITAT POMPEU FABRA

Lección 3

Funcionamiento Singleton

- TestViewController
  - Utiliza la instancia compartida de mySingleton.
  - Llama directamente a los métodos de la instancia compartida de mySingleton sin inicializar objeto alguno desde TestViewController.

- Puede tener su propia lógica de programación independiente de TestViewController.

- Puede combinarse con *Delegate*.

idEC UNIVERSITAT POMPEU FABRA

Lección 3

Ejemplo VIII

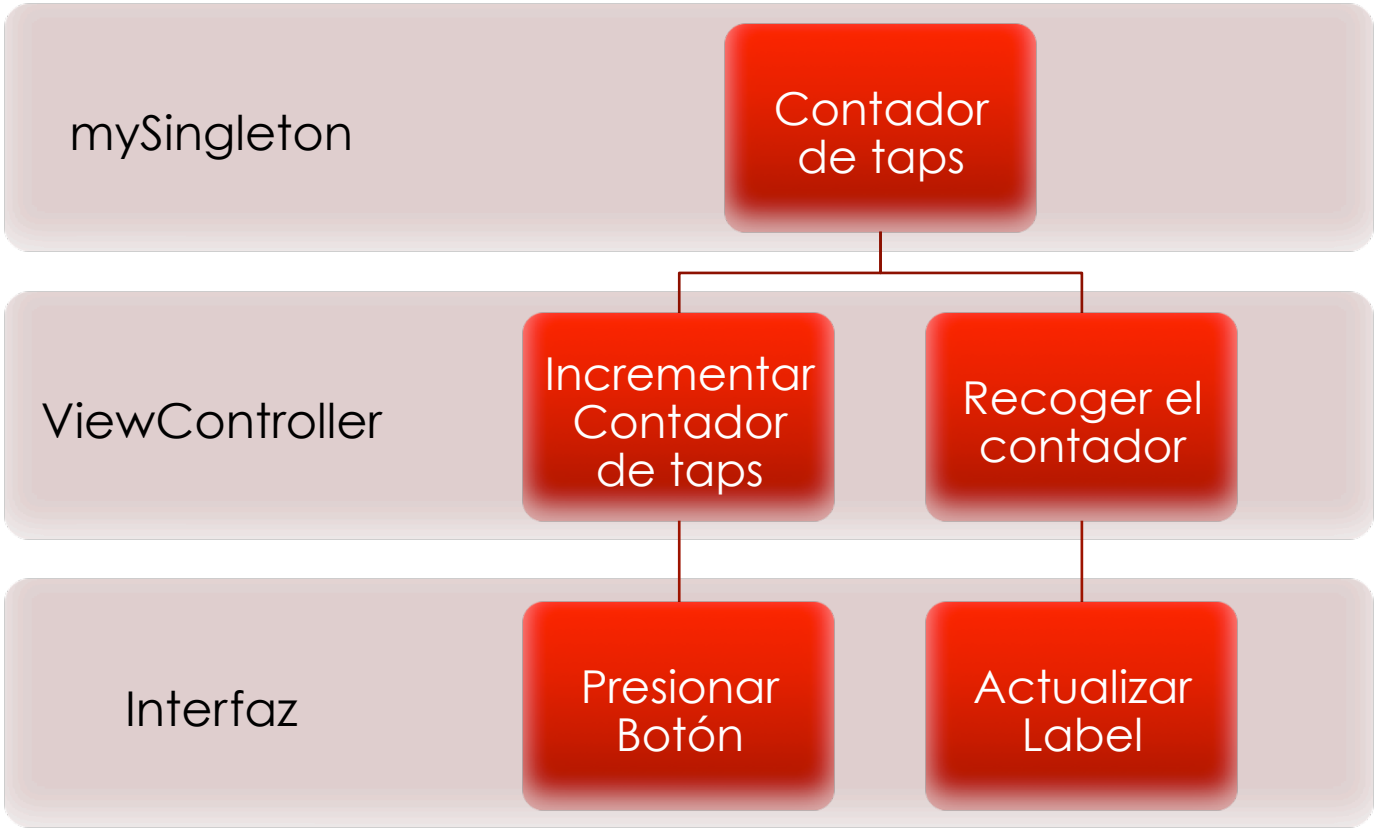
- Aplicación sencilla.
  - Label
  - Botón

- Crear un singleton para almacenar el número de veces que se presiona el botón.
- Una vez presionado el botón e incrementado el contador de taps, actualizar el contenido de la label con el número de veces que se ha presionado el botón.

idEC UNIVERSITAT POMPEU FABRA

Lección 3

Ejemplo VIII Gráficamente



idEC UNIVERSITAT POMPEU FABRA

Lección 3

Ejercicio IV

- Aplicación sencilla.
  - 2 Labels
  - Input
  - 4 Botónes

- Utilizaremos *mySingleton* para almacenar el número de veces que se presiona el botón 1.
- Boton 1 incrementa el contador de taps y lo guarda en *mySingleton*.
- Botón 2 guarda el contenido del input en la variable myText de *mySingleton*.
- Botón 3 reinicia los Labels.
- Botón 4 muestra en Label 1 el contador de taps y en Label 2 el contenido de myText..

Curso de especialización

Programación de Aplicaciones para iPhone/ iPad de Apple

Cocoa: *NSUserDefaults*

Ángel Díaz

angel@blueplastic.net

idEC UNIVERSITAT POMPEU FABRA

Lección 3

NSUserDefaults (I)

- Nos permite almacenar datos de forma "persistente".
- Recomendable para datos "pequeños".
- Persistente a salir y entrar a la aplicación.
- NO persistente a desinstalar y instalar la aplicación.

NSUserDefaults (II)

- Recoger un valor
  - `[[NSUserDefaults standardUserDefaults] valueForKey:@"key"];`

- Setear un valor
  - `[[NSUserDefaults standardUserDefaults] setValue:@"value" forKey:@"key"];`

idEC UNIVERSITAT POMPEU FABRA

Lección 3

Ejemplo IX

- Aplicación sencilla.
  - Label
  - Botón

- Utilizaremos *NSUserDefaults* para almacenar el número de veces que se presiona el botón.
- El Botón incrementa el contador de taps y guarda en *NSUserDefaults*.
- En el Label mostramos el contador de taps.

idEC UNIVERSITAT POMPEU FABRA

Lección 3

Ejercicio V

- Aplicación sencilla.
  - 2 Labels
  - Input
  - 2 Botónes

- Utilizaremos *NSUserDefaults* para almacenar el número de veces que se presiona el botón 1.
- Boton 1 incrementa el contador de taps y guarda en *NSUserDefaults*.
- Botón 2 guarda el contenido del input en *NSUserDefaults.*
- Label 1 mostramos el contador de taps.
- Label 2 mostramos el campo almacenado en *NSUserDefaults.*