



POLITÉCNICA

ETSIT
UPM

dit
UPM

Desarrollo de Apps para iOS Container Views

IWEB,LSWC 2013-2014
Santiago Pavón

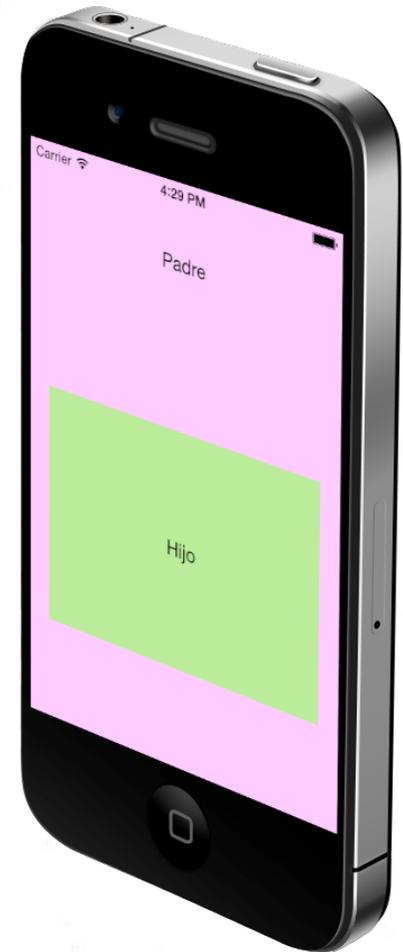
ver: 2014.03.15

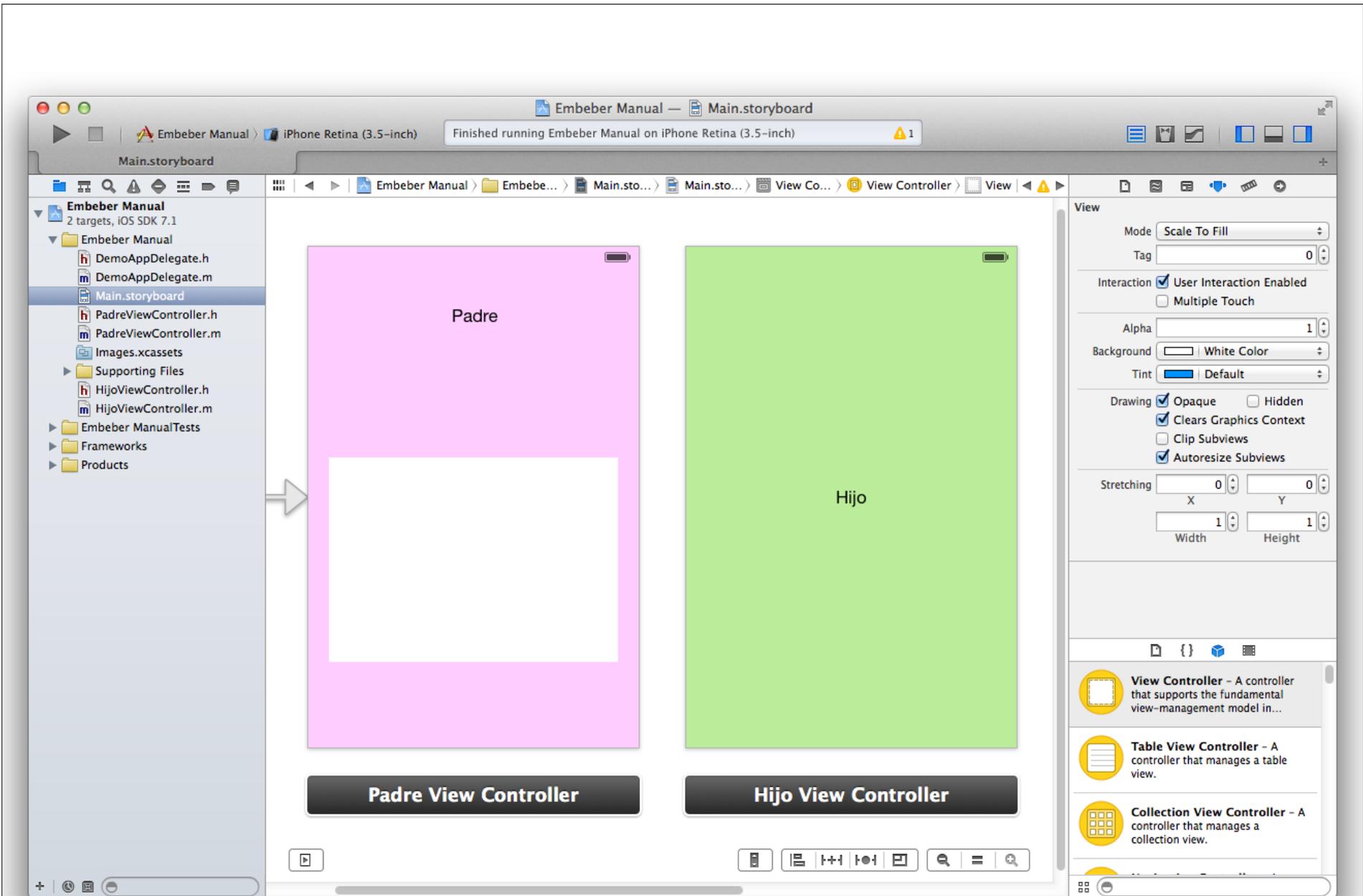
¿En qué Consiste?

- Objetivo:
 - Que un View Controller muestre dentro de su vista la vista de otro View Controller.
 - Es interesante para evitar que el código de un VC sea muy grande. Parte de la funcionalidad se desarrolla en otro VC que luego se incluye en la vista. También para facilitar la reutilización.
- Tareas:
 - Además de construir la jerarquía completa de UIViews, es necesario construir la jerarquía de UINavigationController para que los métodos del ciclo de vida de los VC incluidos se llamen adecuadamente:
 - al rotar el terminal, cuando aparece y desaparece el VC, al recalcular layouts, ...

Programáticamente

- Primero hay que crear la jerarquía de UIViews, que será una mezcla de las vistas de varios VC.
- Segundo hay que crear la jerarquía de VC.
 - Para añadir/eliminar un UIViewController como hijo:
 - addChildViewController:
 - removeFromParentViewController:
 - Antes de añadir o eliminar un VC hijo:
 - willMoveToParentViewController:
 - Este método ya se llama automáticamente desde addChildViewController:
 - Para hacer transiciones entre VC hijos:
 - transitionFromViewController:toViewController:
 - duration:options:animations:completion:
 - Después de añadir o eliminar un VC hijo:
 - didMoveToParentViewController:
 - No se llama automáticamente.
 - Debemos llamarlo después de addChildViewController:
 - y una vez hayan terminado las animaciones de las transiciones.





```

#import "PadreViewController.h"
#import "HijoViewController.h"

@interface PadreViewController ()
@property (weak, nonatomic) IBOutlet UIView *container;
@property (weak, nonatomic) HijoViewController *ivc;
@end

@implementation PadreViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    // Creo el objeto VC hijo:
    HijoViewController * ivc = [self.storyboard
                                instantiateViewControllerWithIdentifier:@"Internal VC"];

    self.ivc = ivc;

    // Construir jerarquia de views
    [self.container addSubview:ivc.view];

    // Construir jerarquia de View Controllers
    [self addChildViewController:ivc];
    [ivc didMoveToParentViewController:self];
}

-(void)viewDidLayoutSubviews {
    // Ajusto geometrias
    self.ivc.view.frame = self.container.bounds;
}

@end

```

Con Storyboard - Interface Builder

- Se usan Container Views
 - Son views especiales que muestran el contenido de otro VC.
- Arrastrarlas desde la biblioteca de objetos hasta el Storyboard.
- Un Container View se enlaza con el View Controller que mostrará con **segues** de tipo **embed**.
 - Con los segues embed también puede sobrescribirse el método **prepareForSegue:sender:** del VC contenedor para configurar el VC embebido.

