



POLITÉCNICA

ETSIT
UPM

dit
UPM

Desarrollo de Apps para iOS

Primera Demo: Hola Mundo

IWEB,LSWC 2013-2014

Santiago Pavón

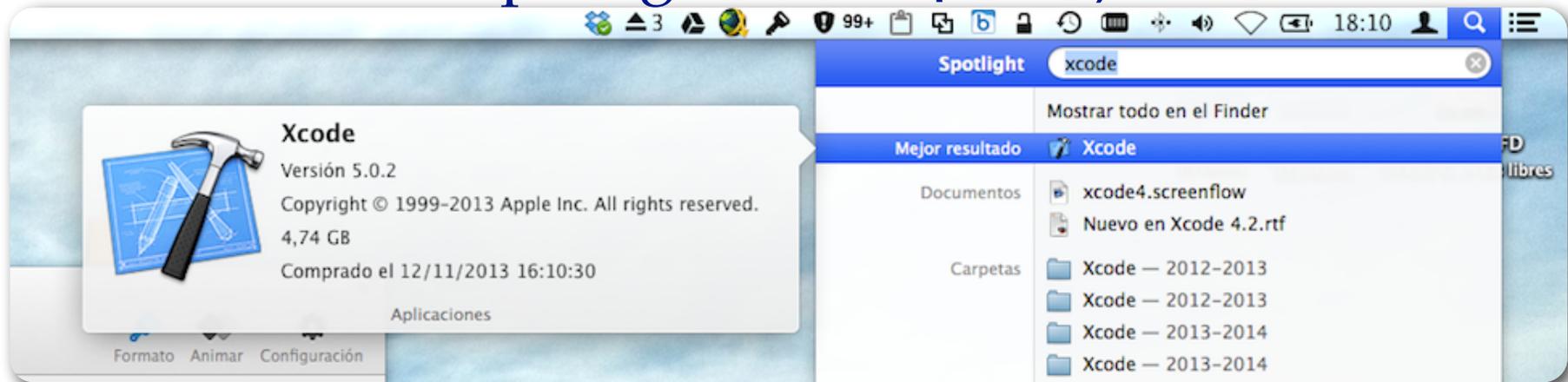
ver: 2014.02.11

Contenido

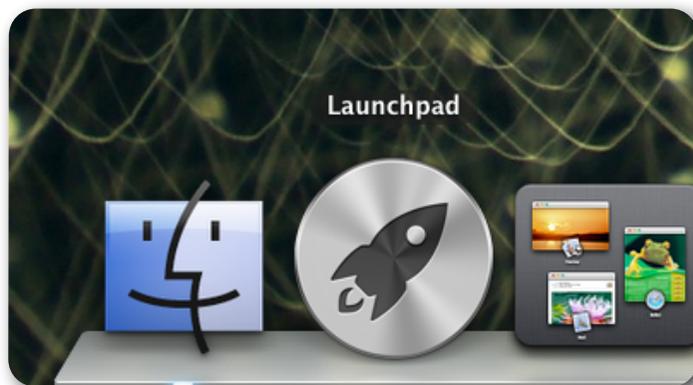
- Familiarizarse con el entorno Xcode:
 - editor, simulador, depurador, ...
- Crear una app sencilla:
 - GUI: botones, etiquetas, mapas, ...
 - Conceptos: IBOutlet, IBAction

Lanzar Xcode

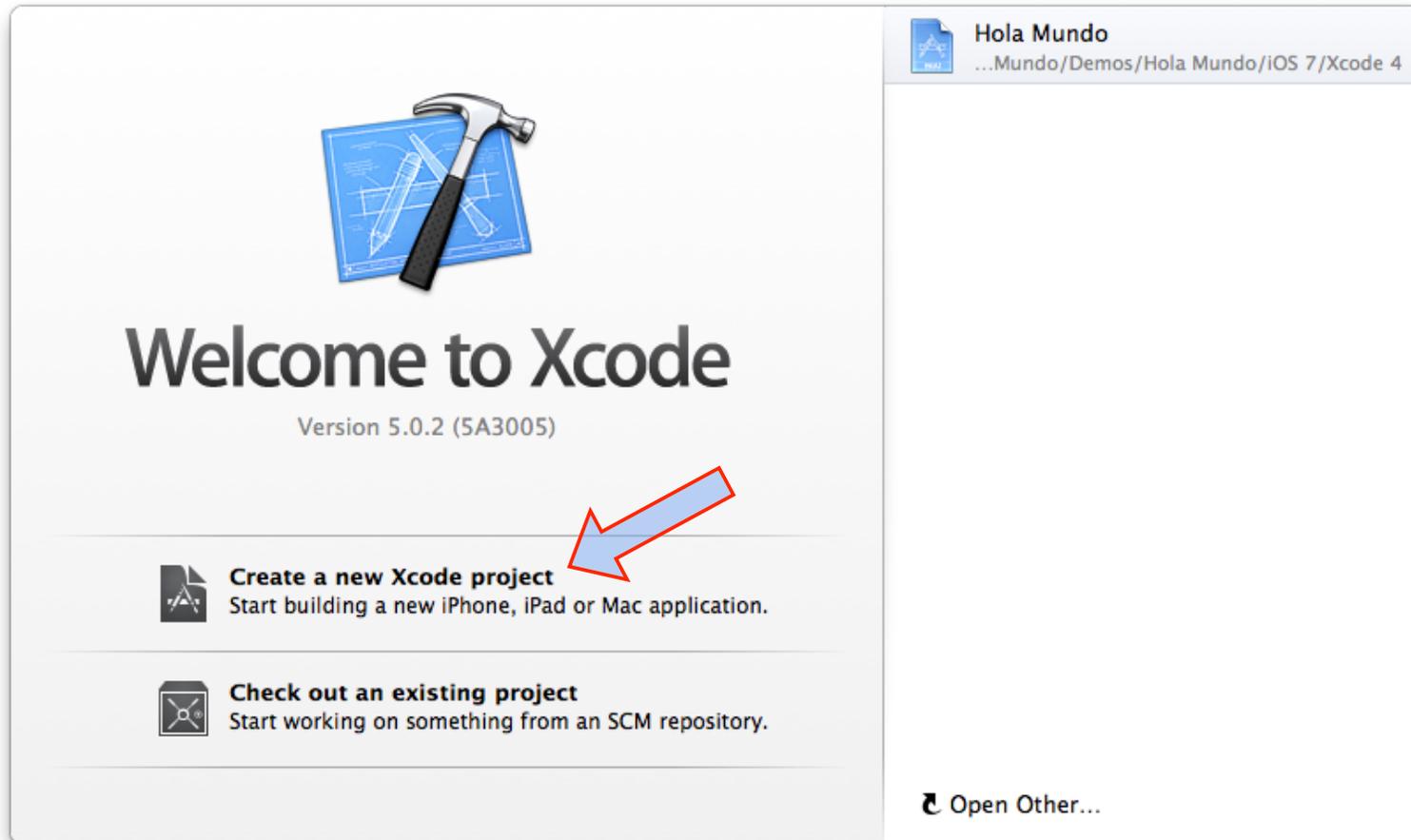
- Buscarlo con Spotlight (⌘-Espacio)



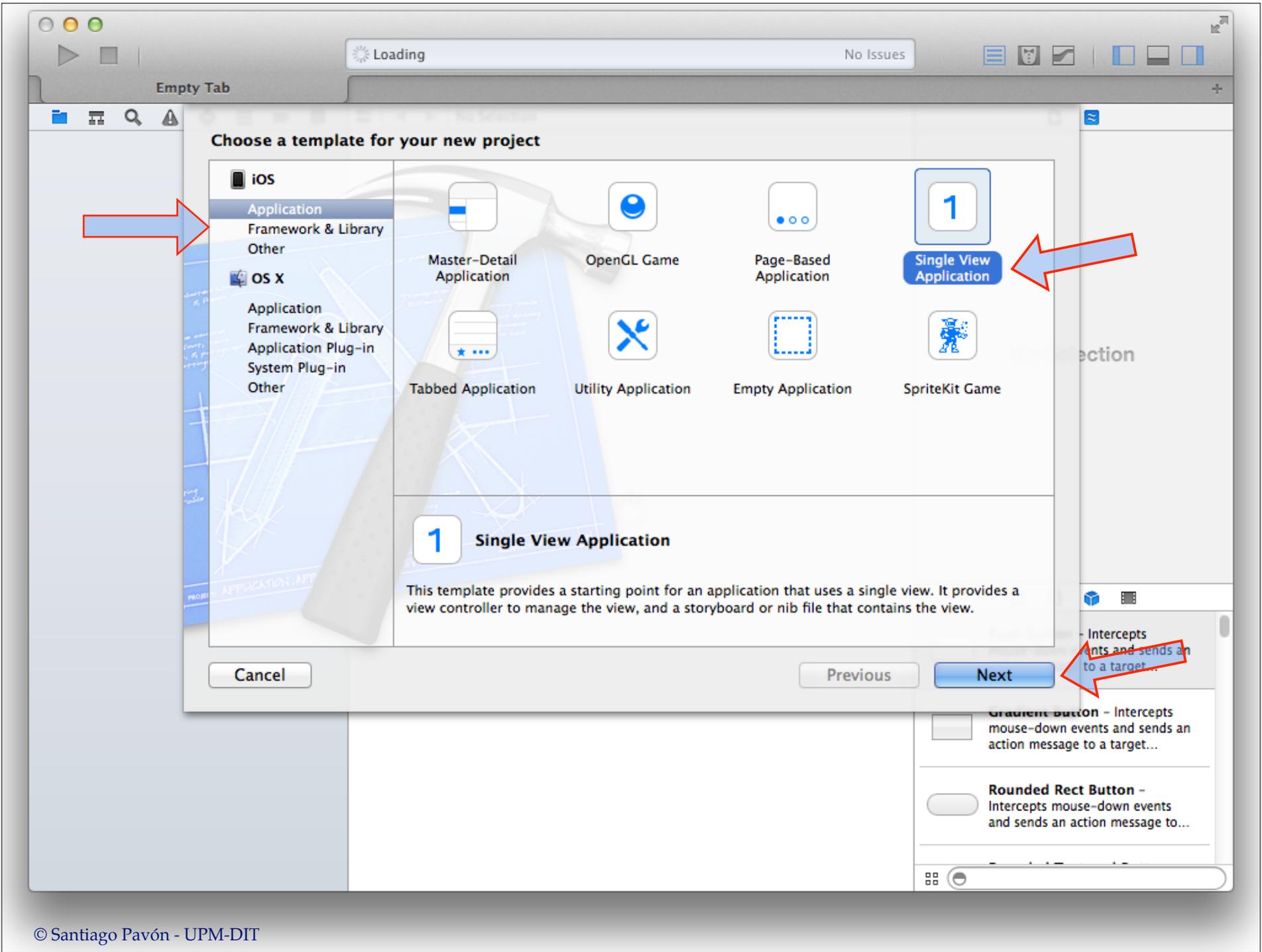
- Buscarlo con Launchpad:

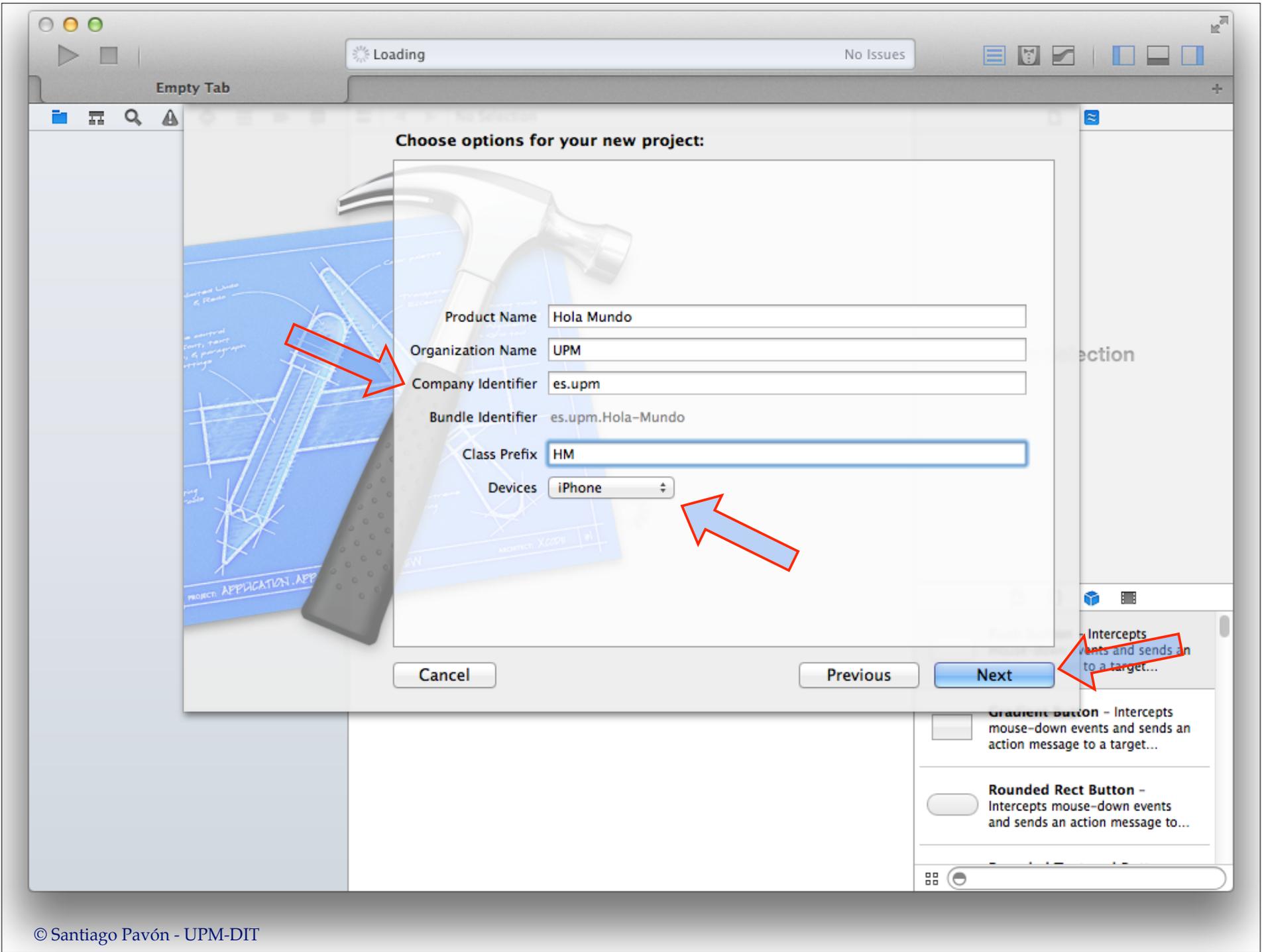


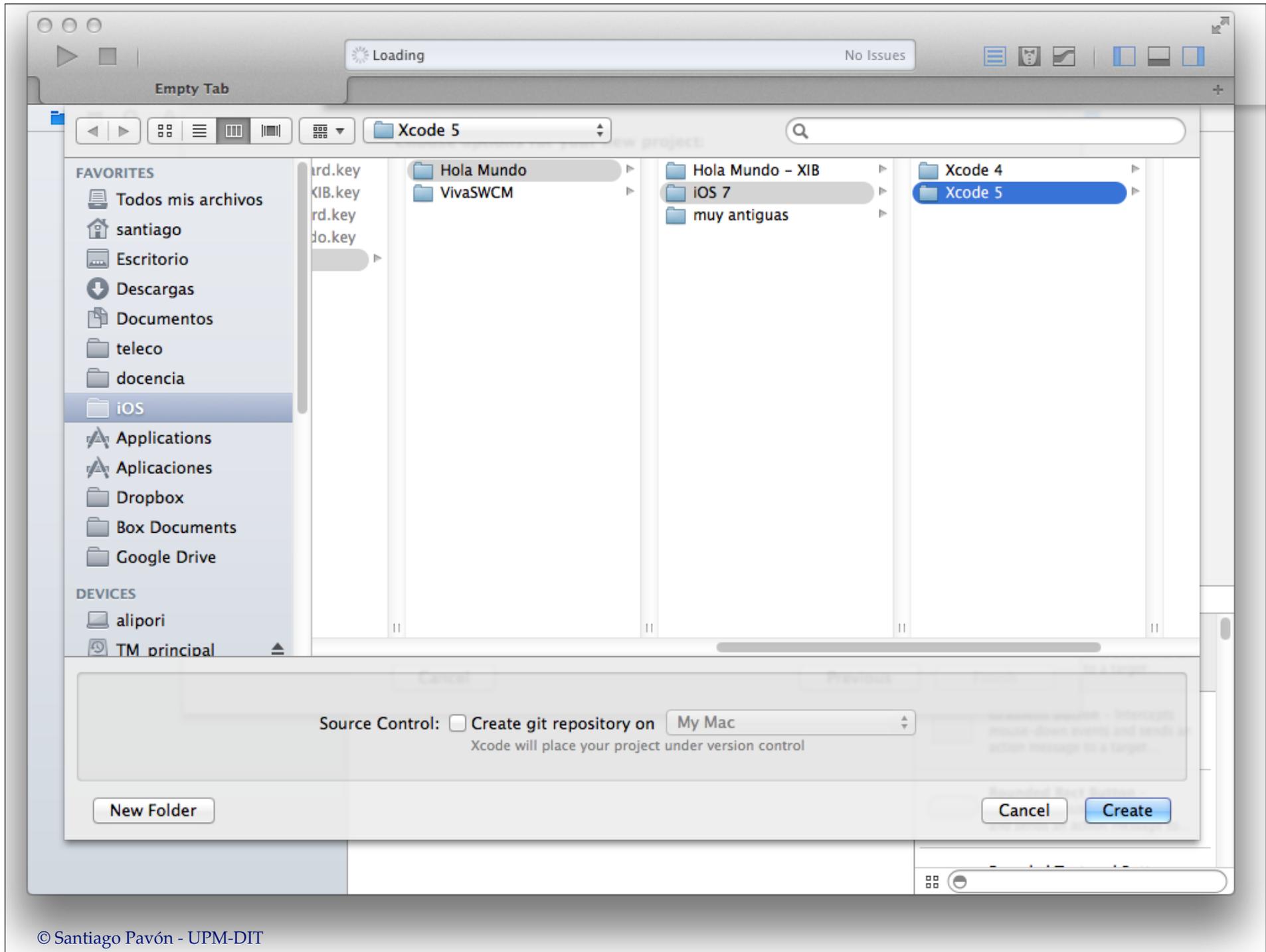
Crear Nuevo Proyecto

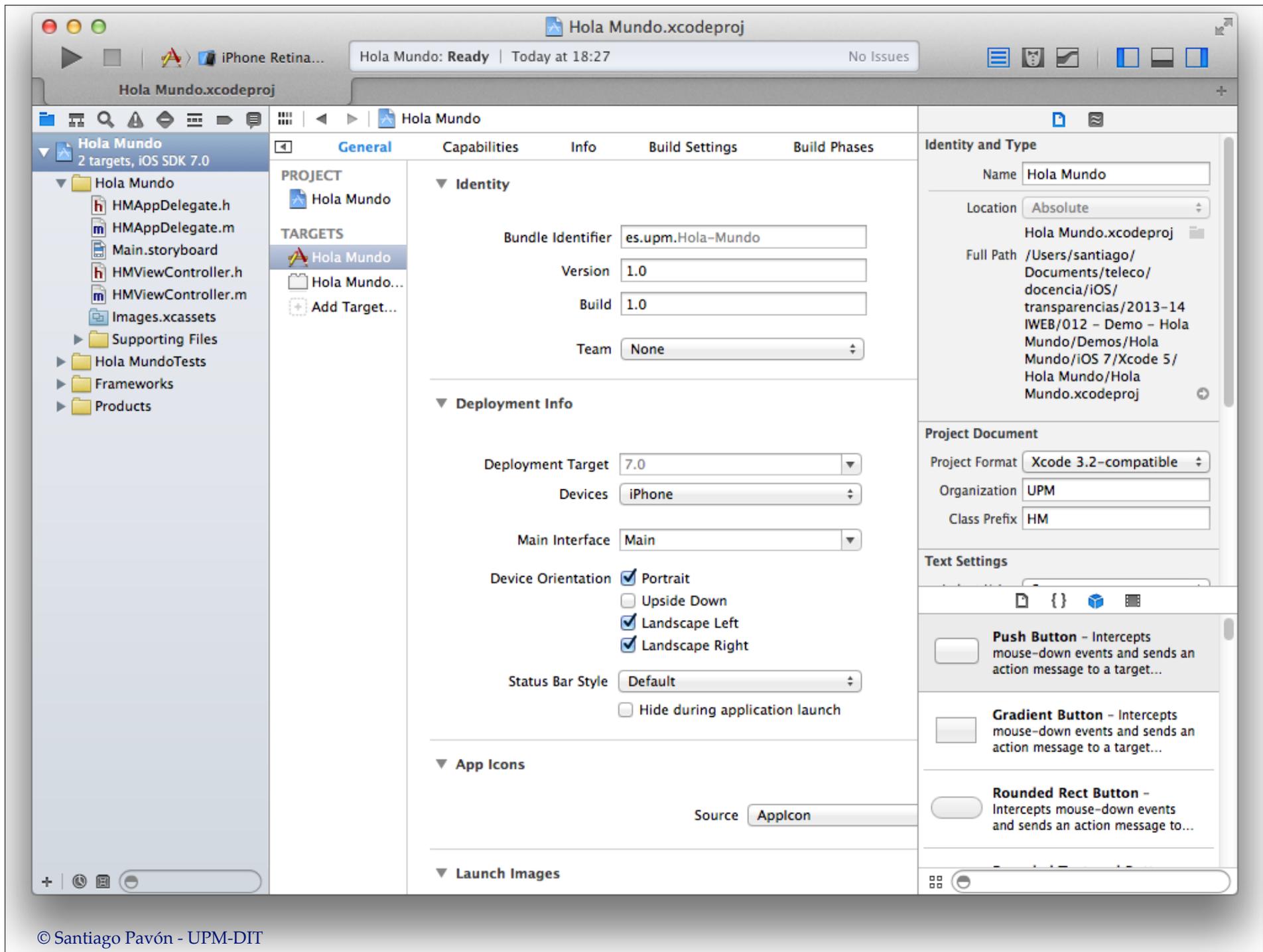


También: **Menú File > New > New Project**



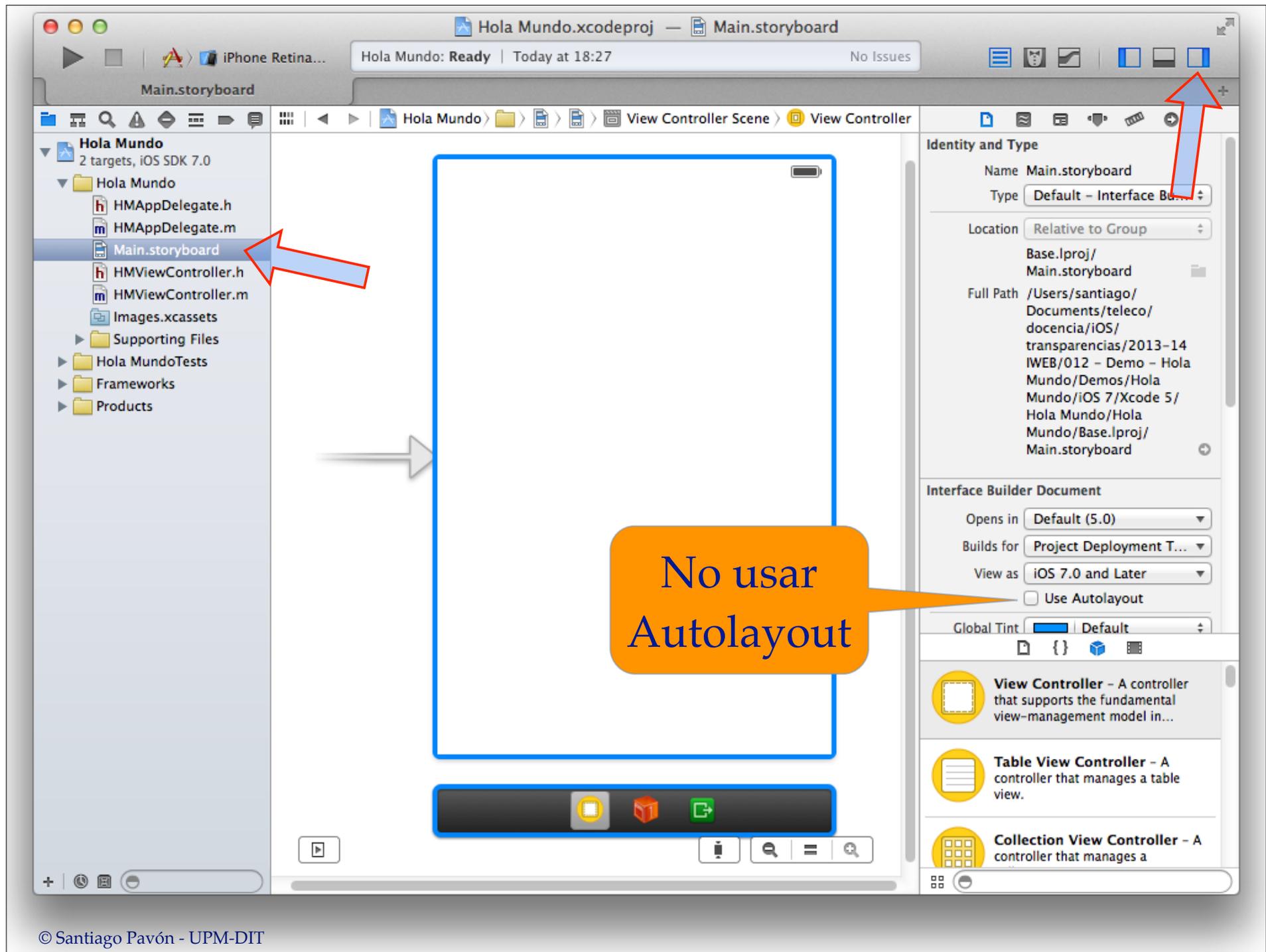


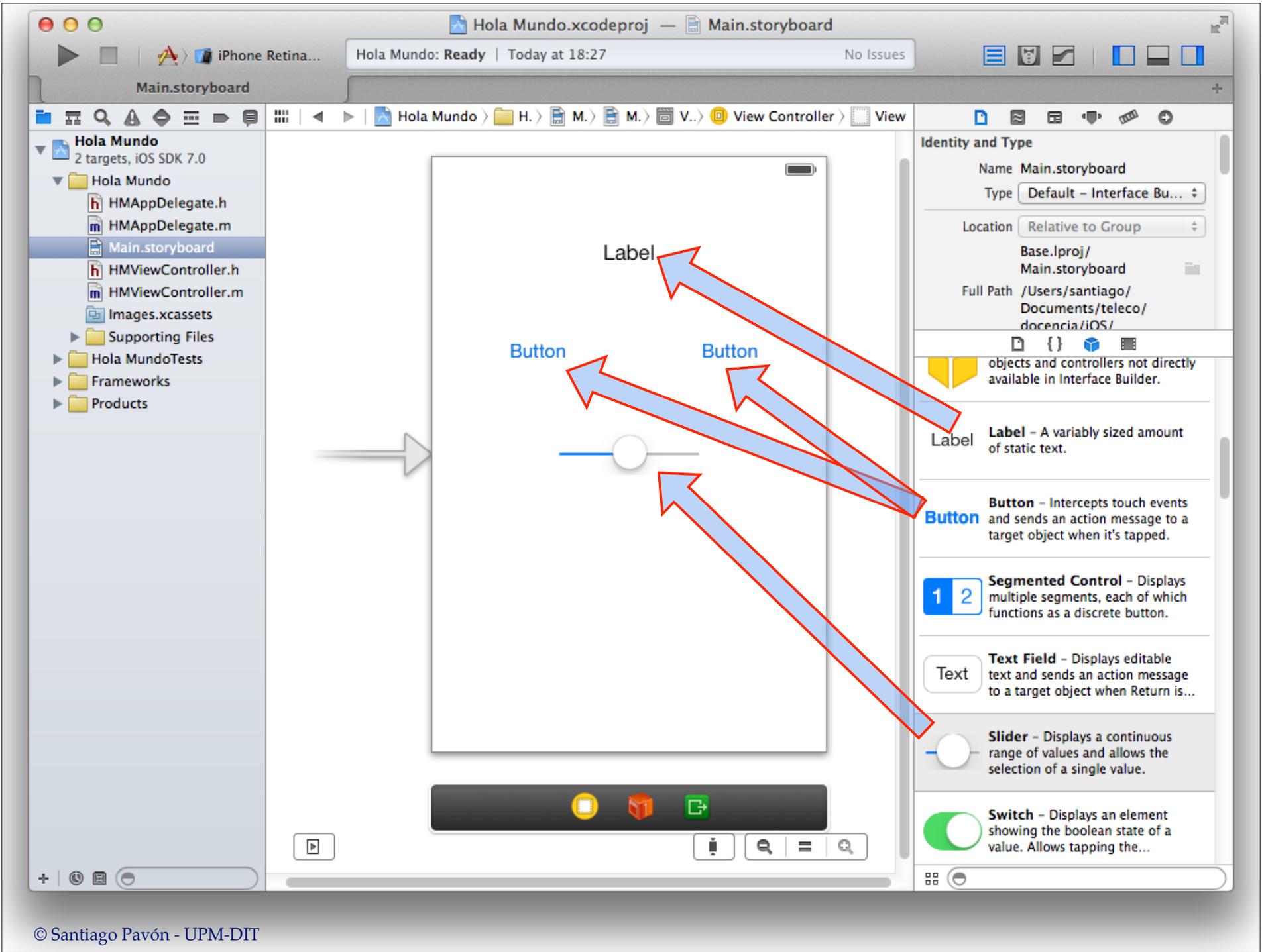




Crear GUI

- Click en el fichero **MainStoryboard.storyboard**
 - Se abre en el editor (Interface Builder)
- Mostrar el inspector.
- No usar Autolayout.
- Desde la librería de objetos arrastrar hasta la vista del ViewController:
 - una UILabel
 - dos UIButton
 - un UISlider





Cambiar atributos

- Seleccionar cada view y cambiar el aspecto usando el inspector

Probarlo

- Compilar y ejecutar



Comportamiento

- Añadiremos a la clase HMViewController:
 - un **IBoutlet** para la Label
 - es una propiedad que apunta a la UILabel creada.
 - dos **IBactions** para atender pulsaciones de los botones
 - son los métodos que se ejecutarán al pulsar los botones.
 - una **IBaction** para atender cambios en el Slider.
 - es el método que se ejecutará al mover el slider.

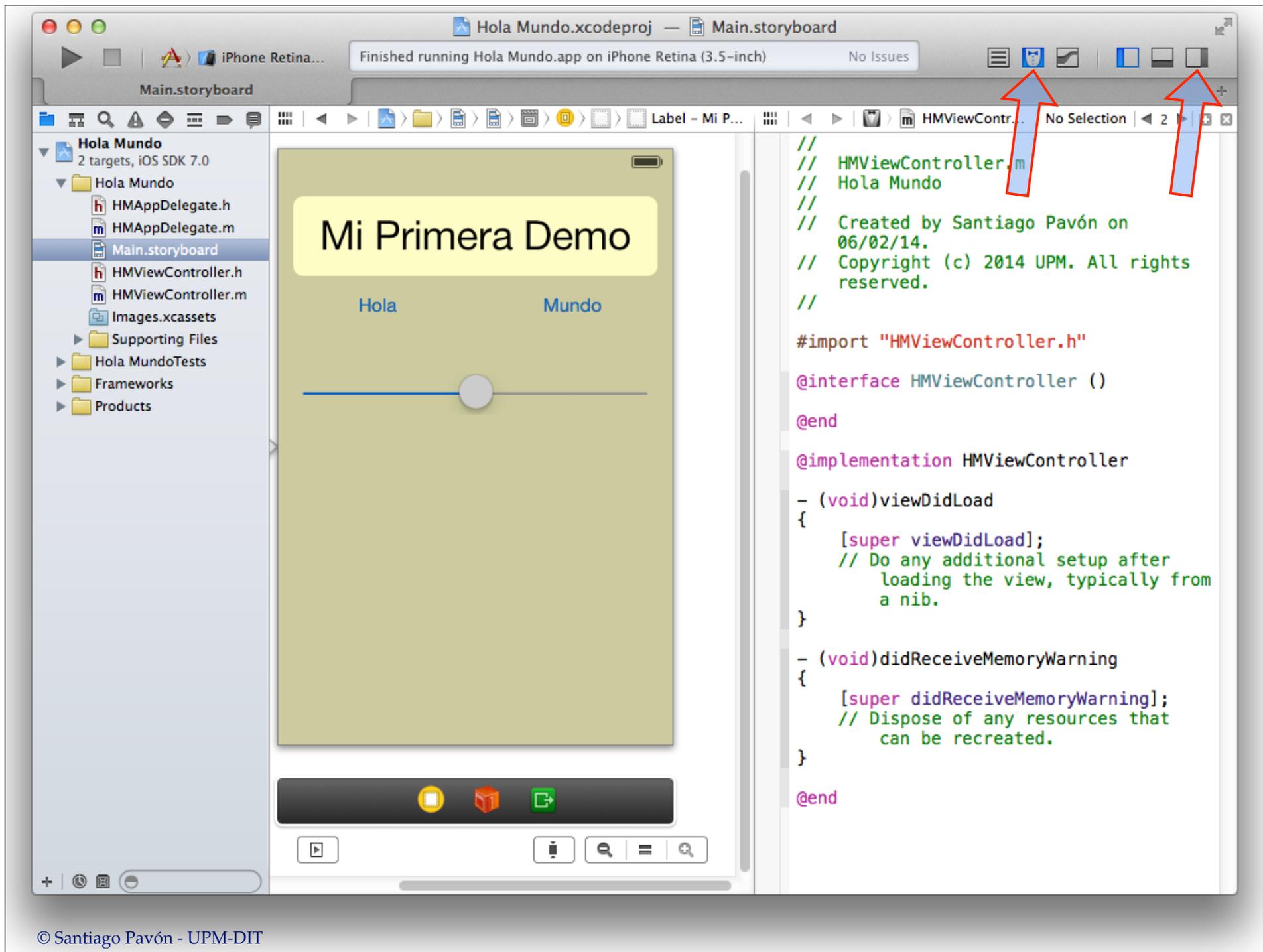
¿Qué es IBOutlet?

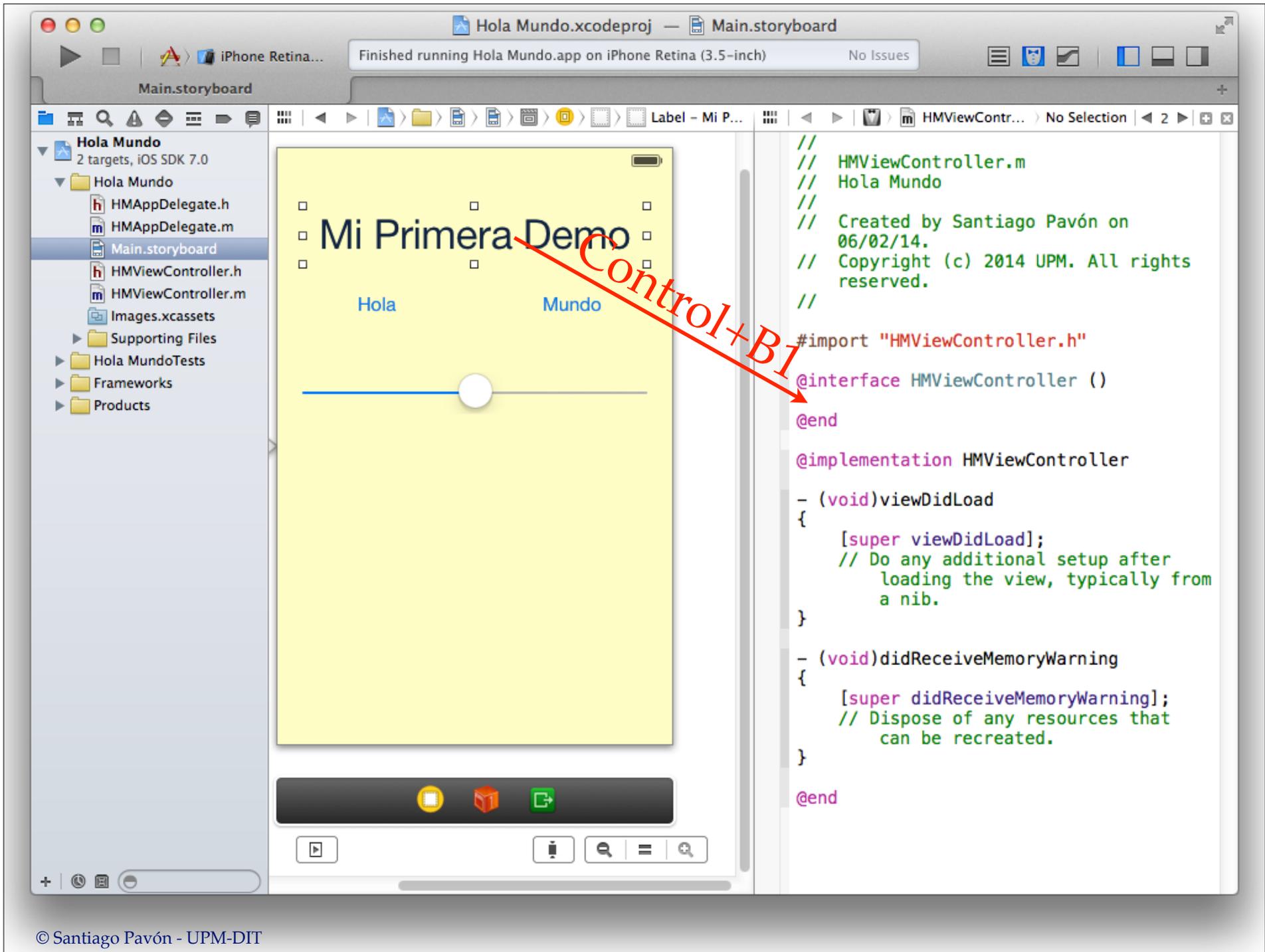
- Es una marca que reconoce Interface Builder
 - Indica que la propiedad marcada con IBOutlet apuntará a un objeto que se ha creado con el Interface Builder.
 - No vamos a crear el objeto mediante código.
- Usaremos Interface Builder para conectar la propiedad y el objeto.
- Es un define a vacío.

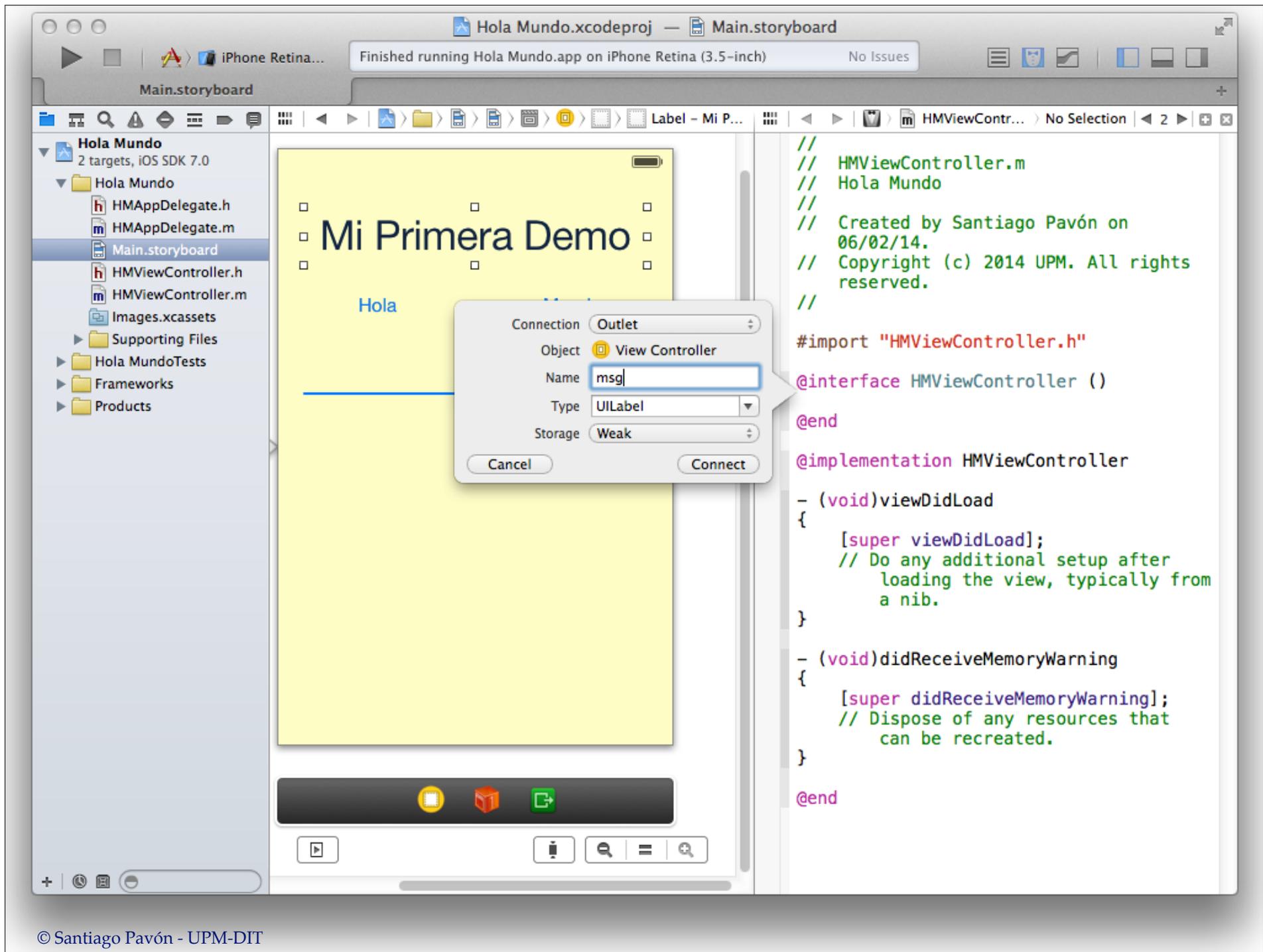
¿Qué es IBAction?

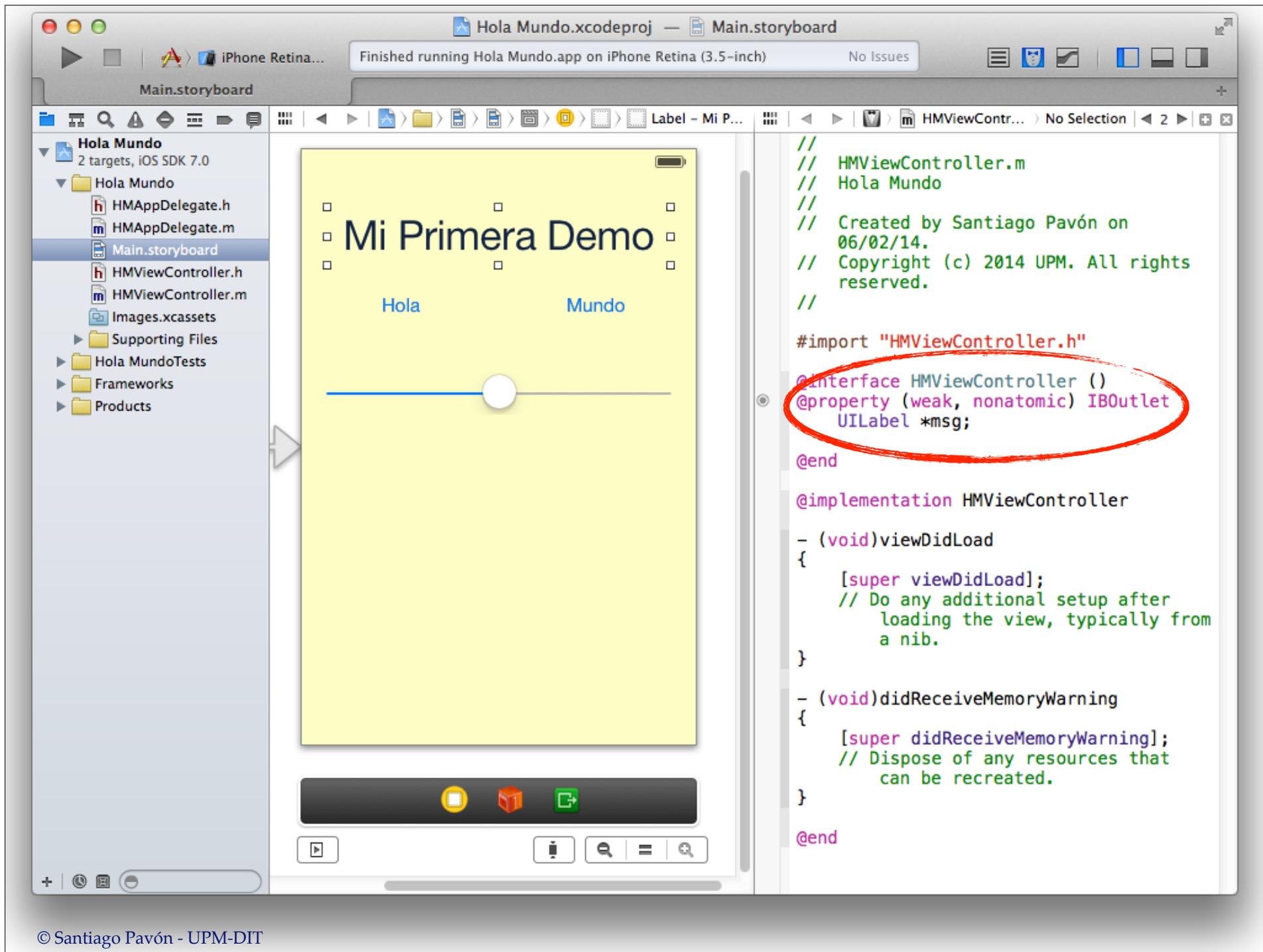
- Es una marca que usa el Interface Builder
 - Indica que el método marcado con IBAction es el target de algún control creado con Interface Builder.
 - No especificaremos el target con código.
- Usaremos Interface Builder para conectar el control y el método.
- Es un define a void.

Crear IBOutlet para la Etiqueta





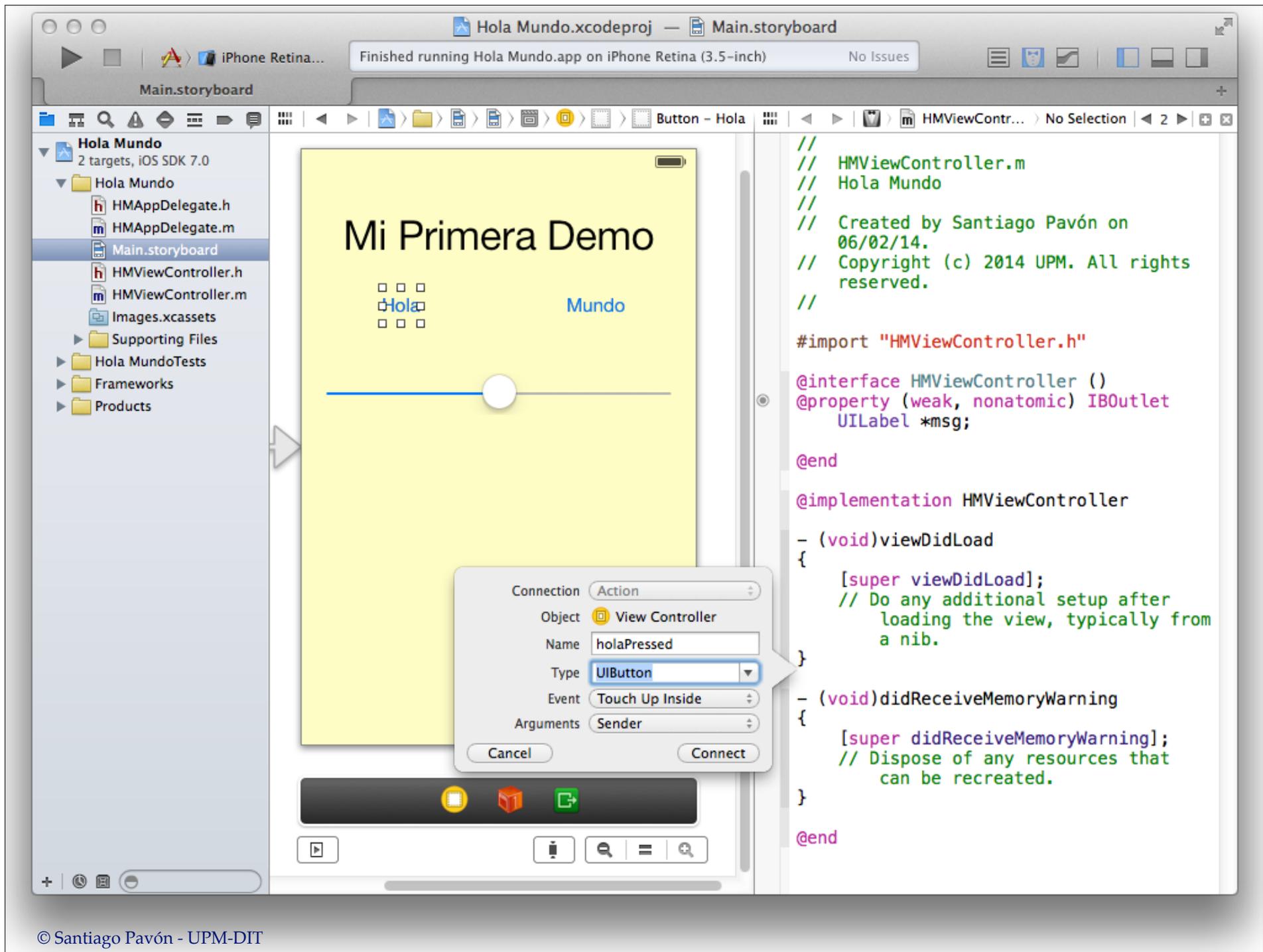


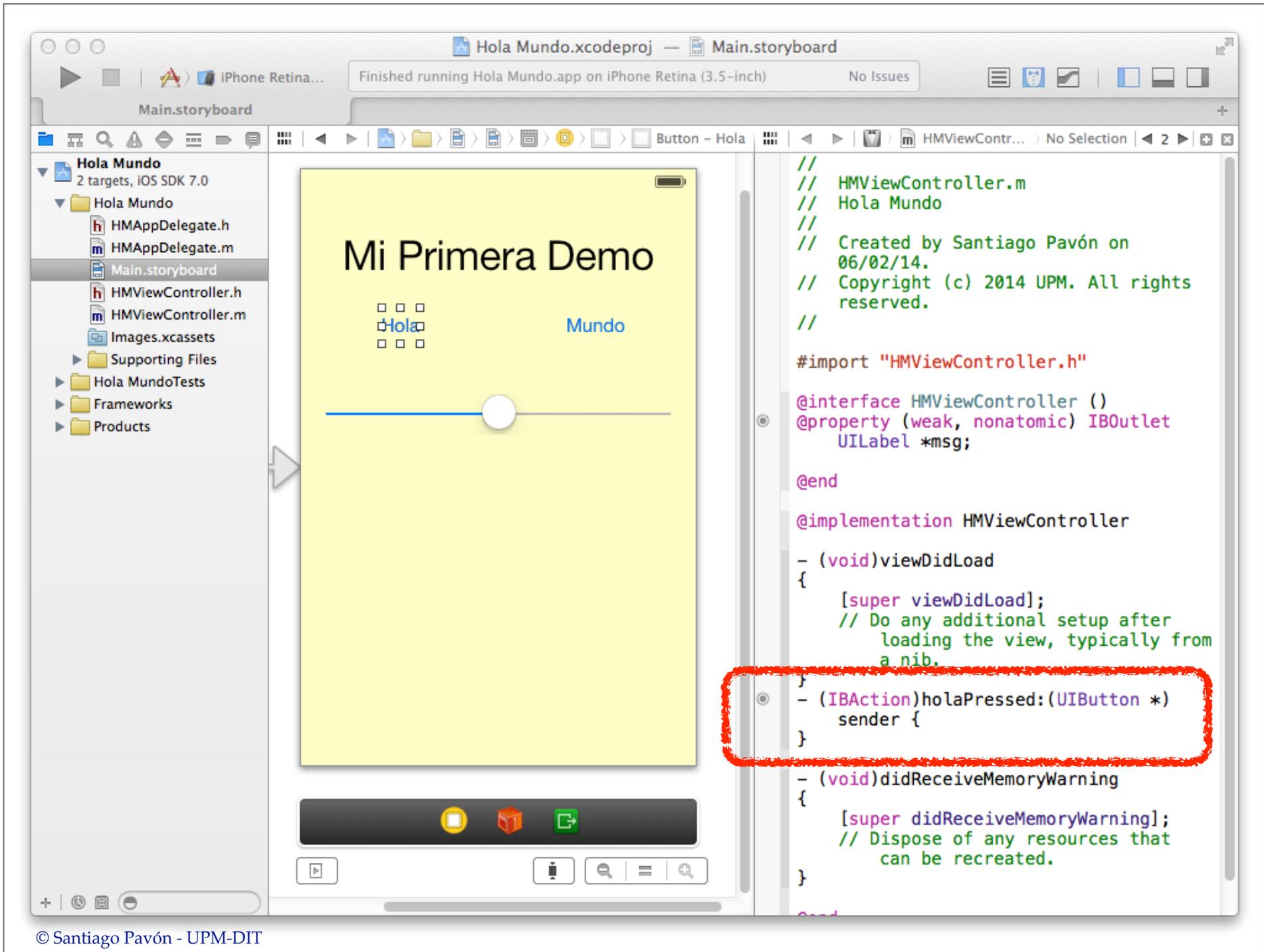


Crear IBActions para los Controles

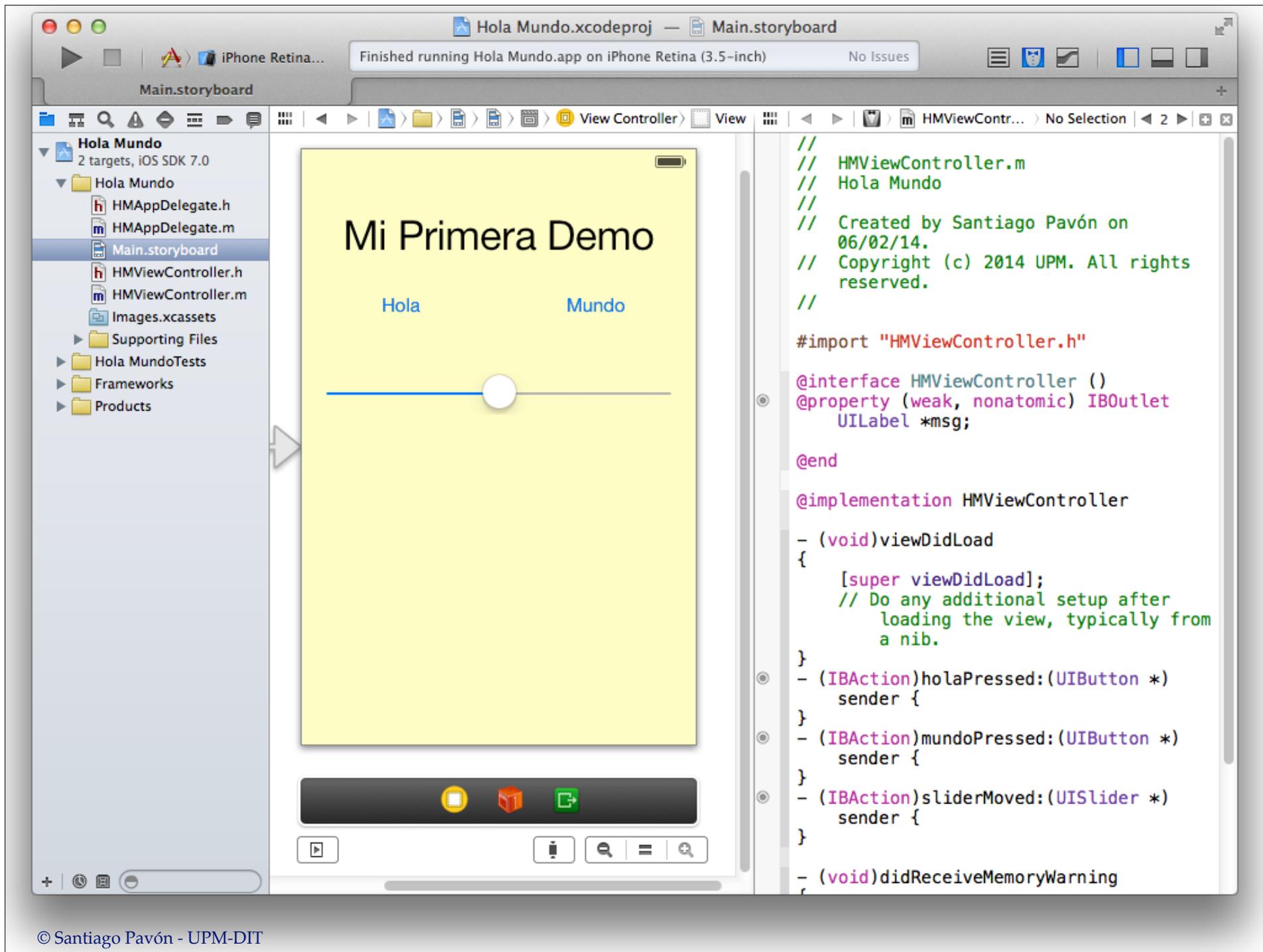
The image shows the Xcode IDE interface. On the left is the Project Navigator showing the project structure for 'Hola Mundo'. The center pane displays a storyboard with a yellow background, the text 'Mi Primera Demo', and a button labeled 'Mundo'. A red arrow points from the text 'Control+B1' to the code editor on the right. The code editor shows the implementation of HMViewController.m, including the viewDidLoad and didReceiveMemoryWarning methods.

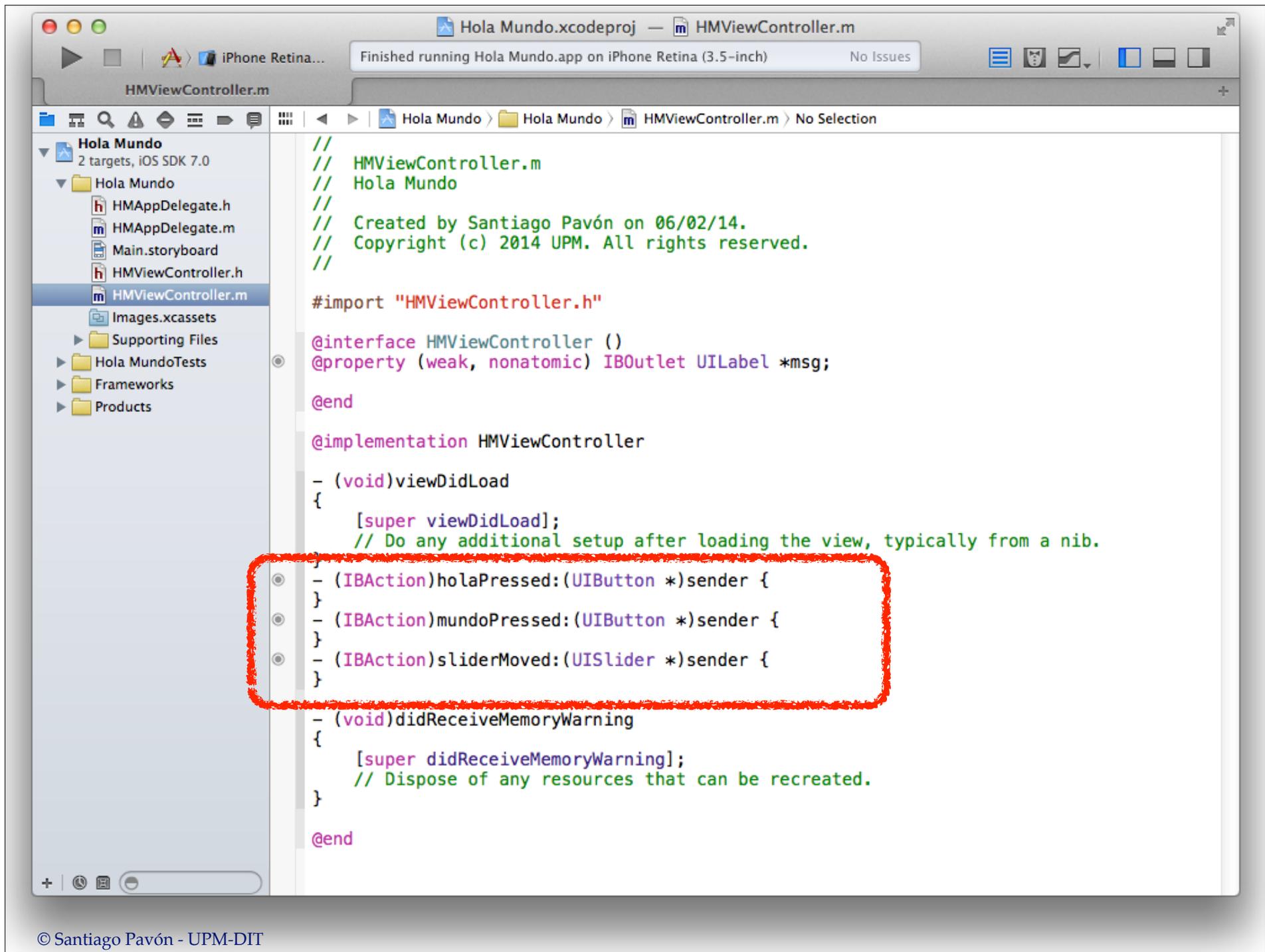
```
//  
// HMViewController.m  
// Hola Mundo  
//  
// Created by Santiago Pavón on  
// 06/02/14.  
// Copyright (c) 2014 UPM. All rights  
// reserved.  
//  
#import "HMViewController.h"  
  
@interface HMViewController ()  
@property (weak, nonatomic) IBOutlet  
    UILabel *msg;  
  
@end  
  
@implementation HMViewController  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after  
    // loading the view, typically from  
    // a nib.  
}  
  
- (void)didReceiveMemoryWarning  
{  
    [super didReceiveMemoryWarning];  
    // Dispose of any resources that  
    // can be recreated.  
}  
  
@end
```





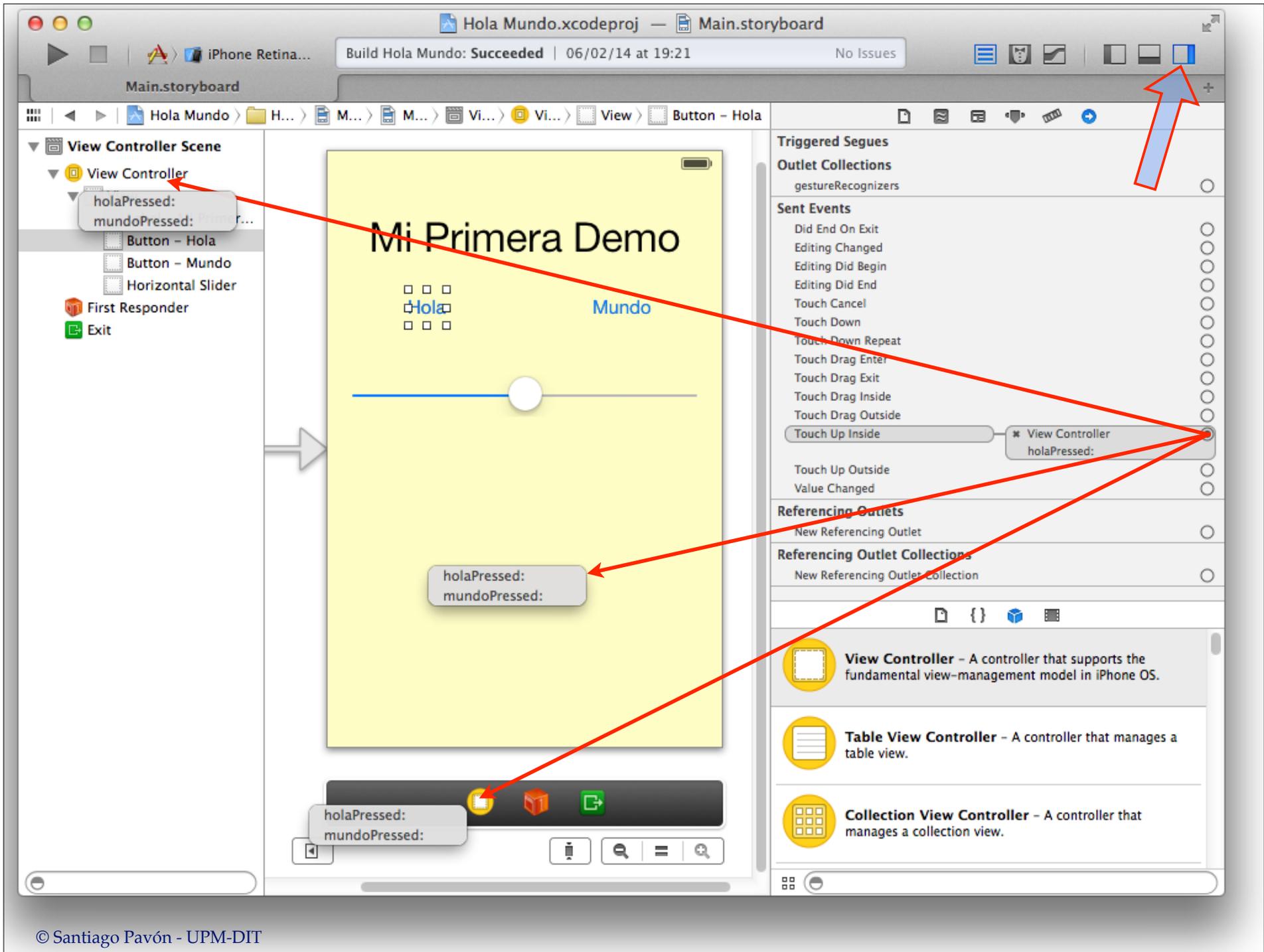
- Repetimos los mismo con el botón mundo y con el slider.





Conectar IBOutlet y IBAction

- Existen varias formas de conectar los Outlets y Actions:
 - En las transparencias anteriores hemos conectado un outlet usando Ctrl+Boton1.
 - Desde un elemento del IB hasta el @interface del fichero .m.
 - Se crea una propiedad.
 - En las transparencias anteriores hemos conectado una action usando Ctrl+Boton1.
 - Desde un control del IB hasta el @implementation del fichero .m.
 - Se crea un método.
- Otras formas de conectar:
 - Seleccionando algún control en el IB, y conectando los eventos desde el inspector de conexiones.
 - Usando el modo asistente, desde el editor de código de los ficheros .h o .m hasta elementos en el editor IB.
 - Usando los círculos situados en margen izquierdo del editor de código.
 - Desde los menús popup que muestran los elementos en el editor IB.
 - Pulsando Control y arrastrando botón 1 del ratón entre elementos en el editor IB.
 - etc...



The screenshot shows the Xcode IDE with the 'Main.storyboard' open. The storyboard displays a yellow background with the text 'Mi Primera Demo' at the top. Below the text are three UI elements: a button labeled 'Hola', a button labeled 'Mundo', and a slider. The code editor on the right shows the implementation of the 'HMViewController' class. The code includes a header file import, an interface declaration for 'HMViewController' with an 'IBOutlet UILabel *msg' property, and an implementation with three actions: 'holaPressed:', 'mundoPressed:', and 'sliderMoved:'. A red arrow points from the 'IBAction' annotations in the code to the corresponding UI elements in the storyboard. Another red arrow points from the 'IBAction' annotations to the 'IBAction' property in the code. A third red arrow points from the 'IBAction' annotations to the 'IBAction' property in the code. A blue arrow points from the 'IBAction' property in the code to the 'IBAction' property in the code.

```
// hola-mundo
// Created by Santiago Pavón on
// 06/02/14.
// Copyright (c) 2014 UPM. All rights
// reserved.

#import "HMViewController.h"

@interface HMViewController ()
@property (weak, nonatomic) IBOutlet
    UILabel *msg;
@end

@implementation HMViewController

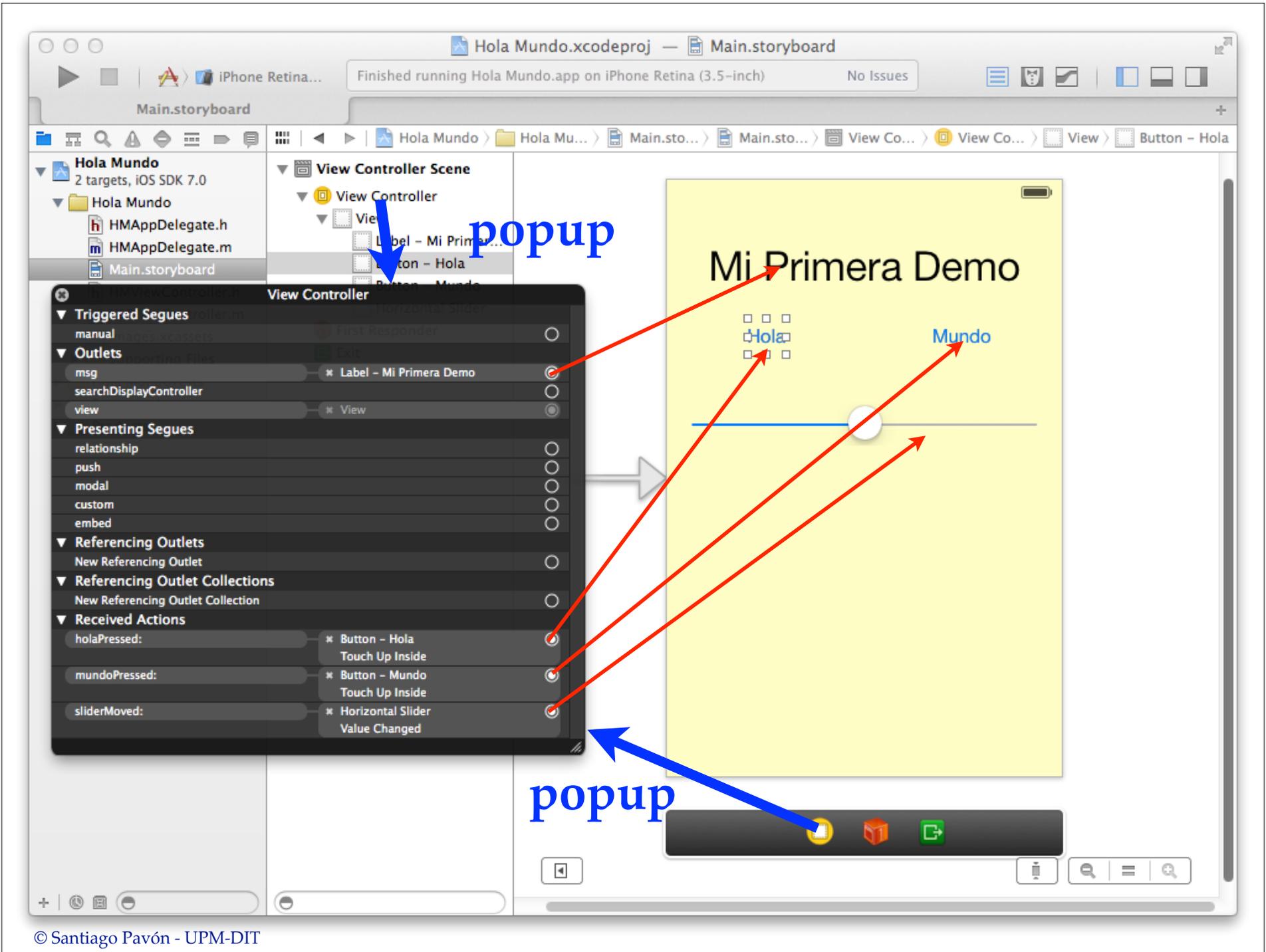
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after
    // loading the view, typically
    // from a nib.
}

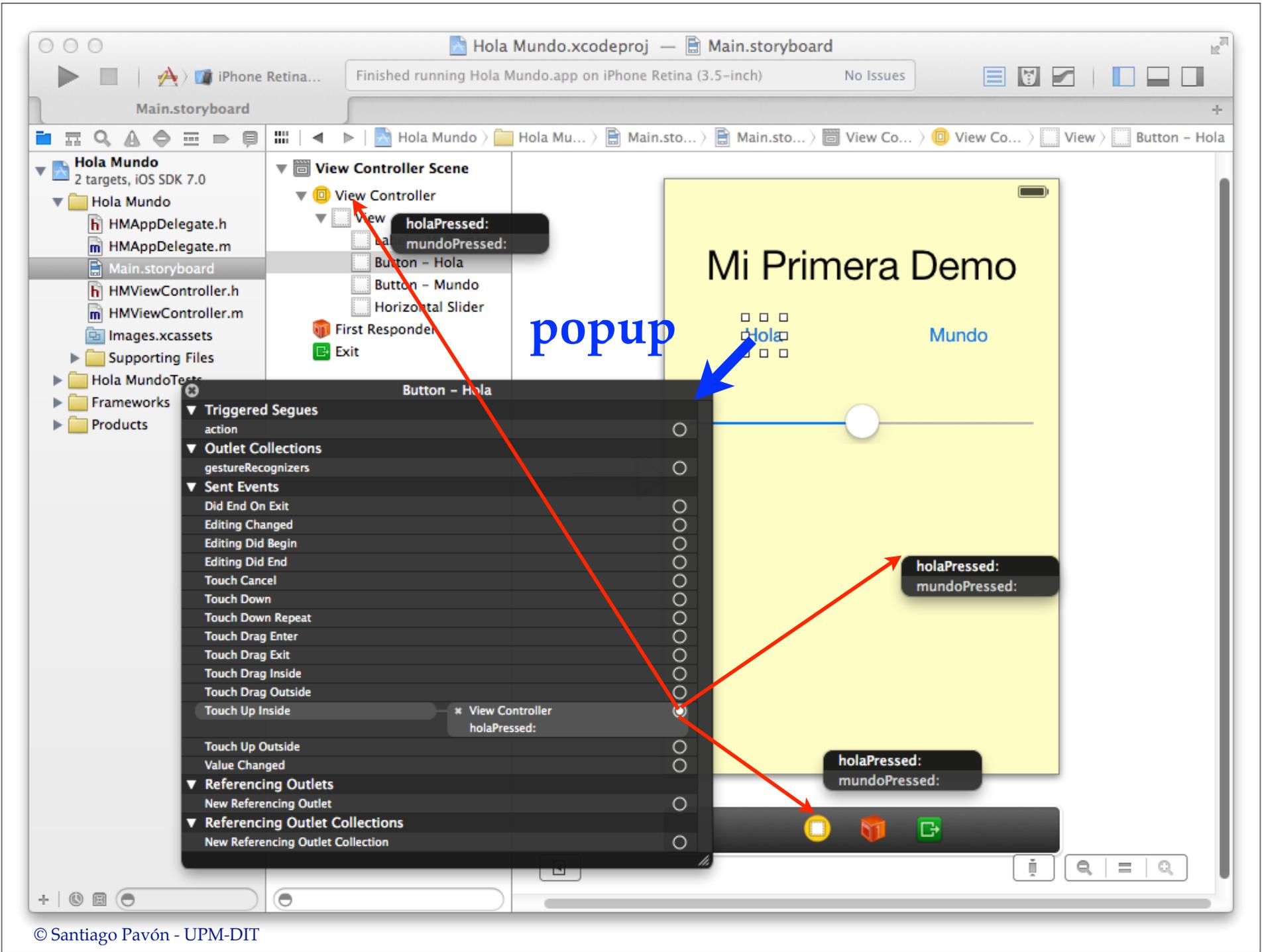
- (IBAction)holaPressed:(UIButton *)
    sender {
}

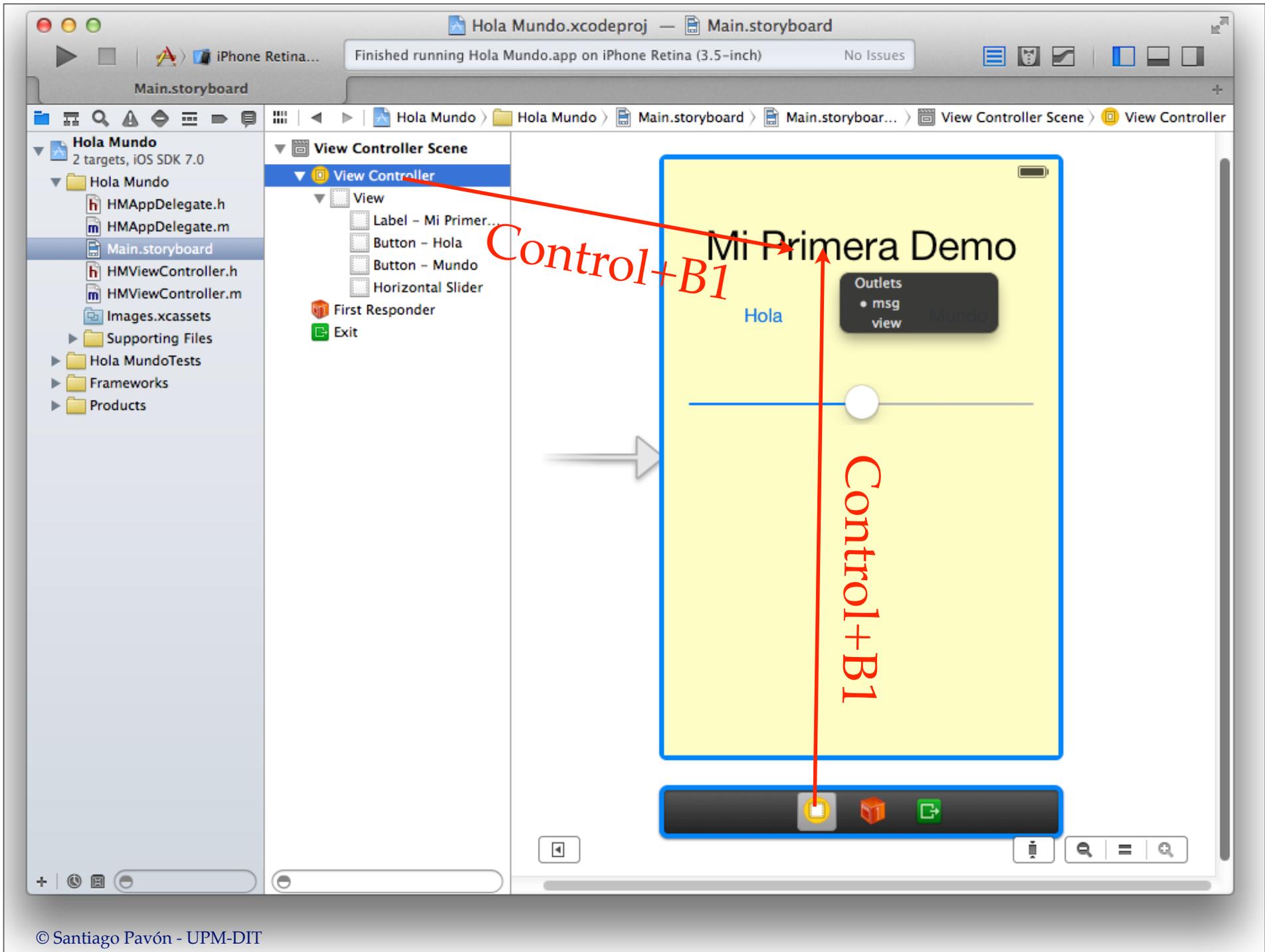
- (IBAction)mundoPressed:(UIButton *)
    sender {
}

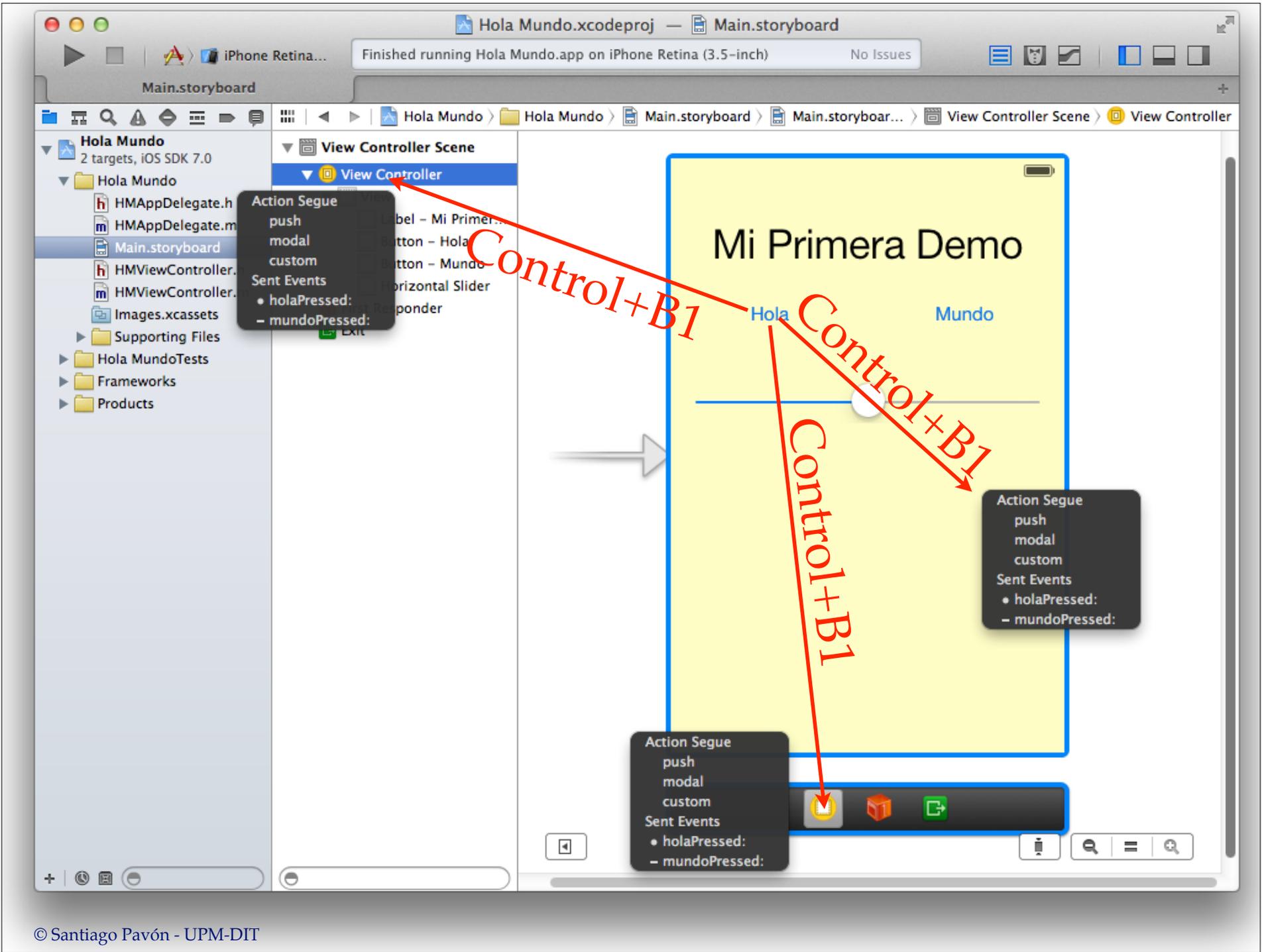
- (IBAction)sliderMoved:(UISlider *)
    sender {
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that
```

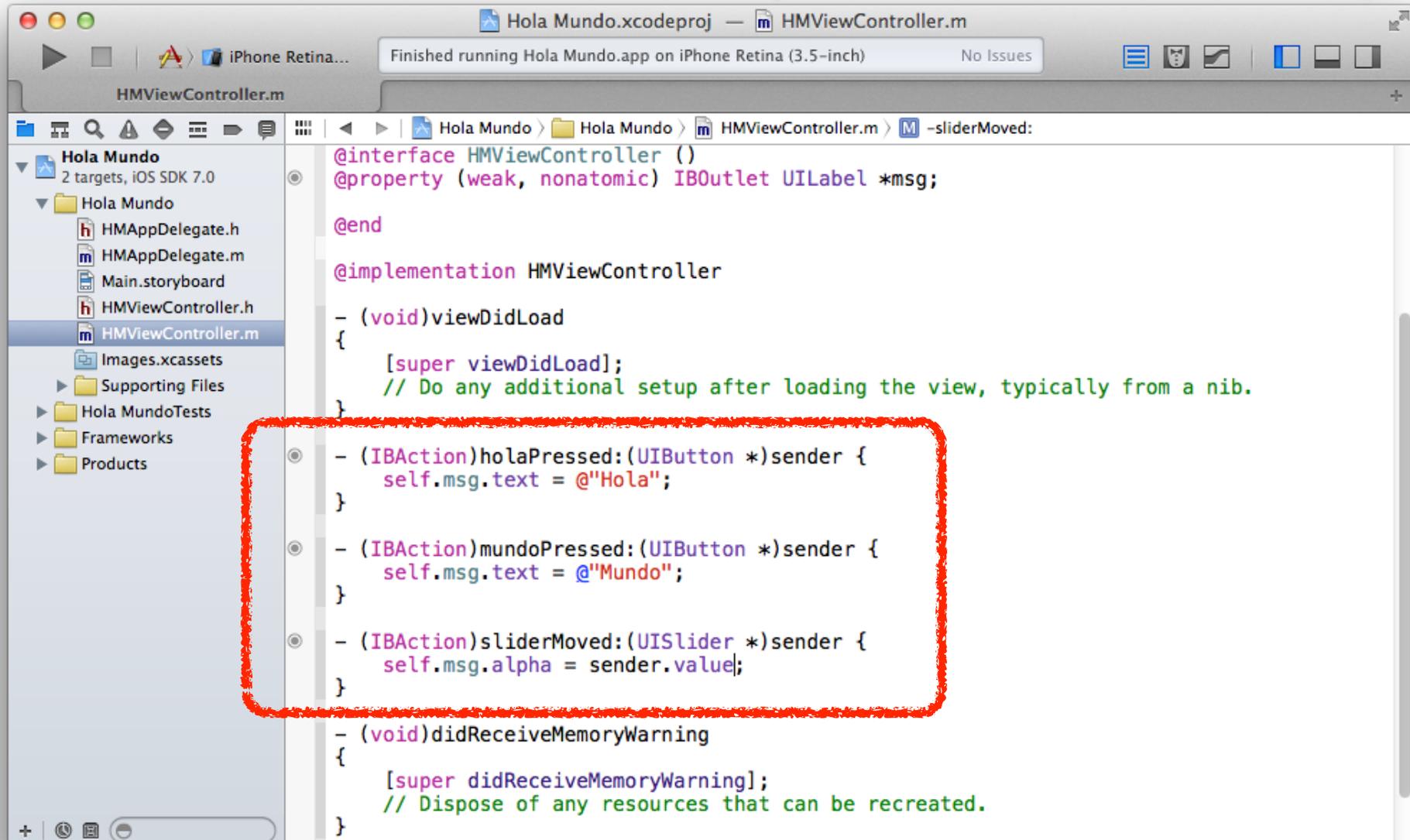








El Código de los Métodos



```

@interface HMViewController ()
@property (weak, nonatomic) IBOutlet UILabel *msg;
@end

@implementation HMViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (IBAction)holaPressed:(UIButton *)sender {
    self.msg.text = @"Hola";
}

- (IBAction)mundoPressed:(UIButton *)sender {
    self.msg.text = @"Mundo";
}

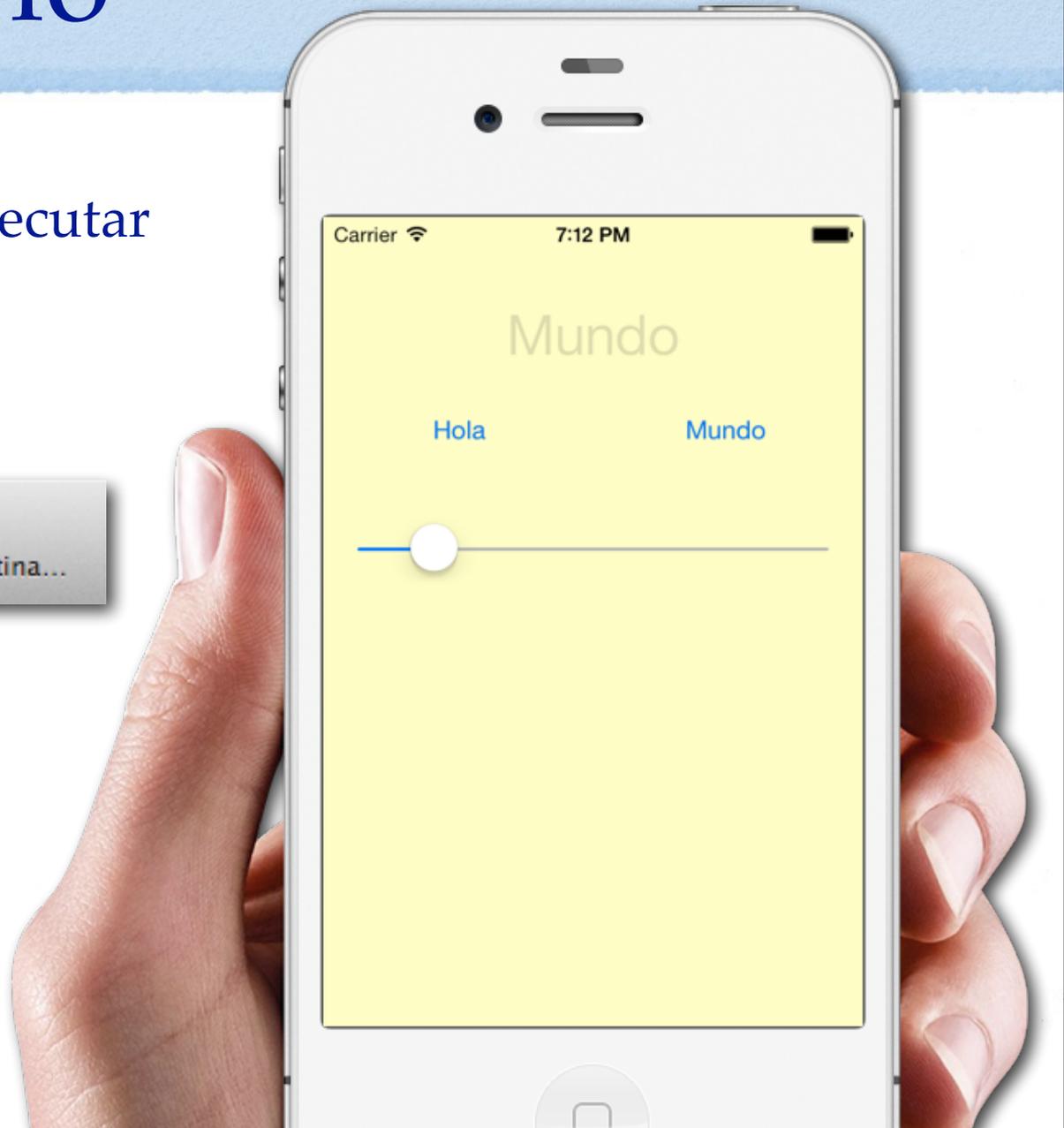
- (IBAction)sliderMoved:(UISlider *)sender {
    self.msg.alpha = sender.value;
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

```

Probarlo

- Salvar, compilar y ejecutar



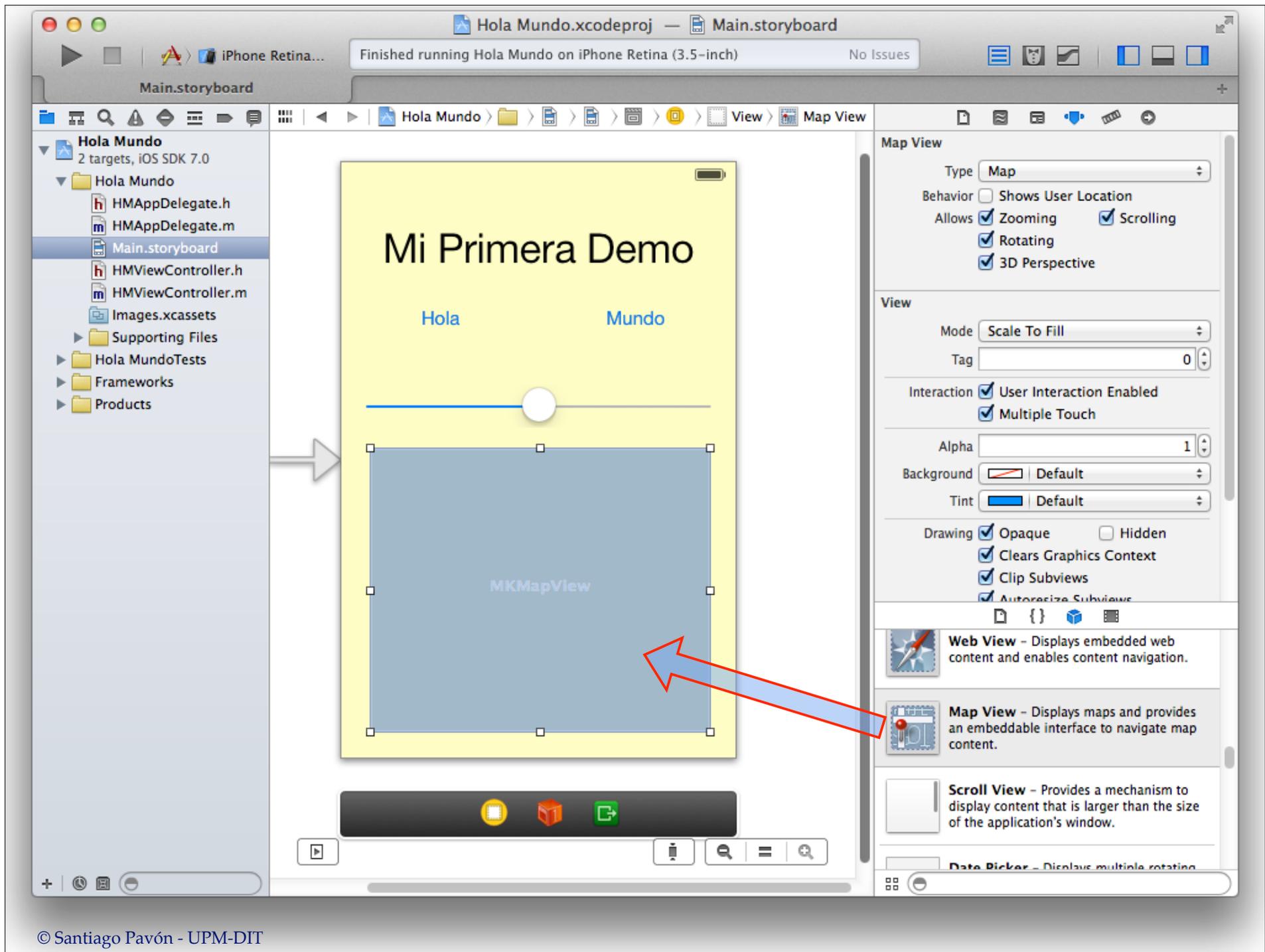
MKMapView

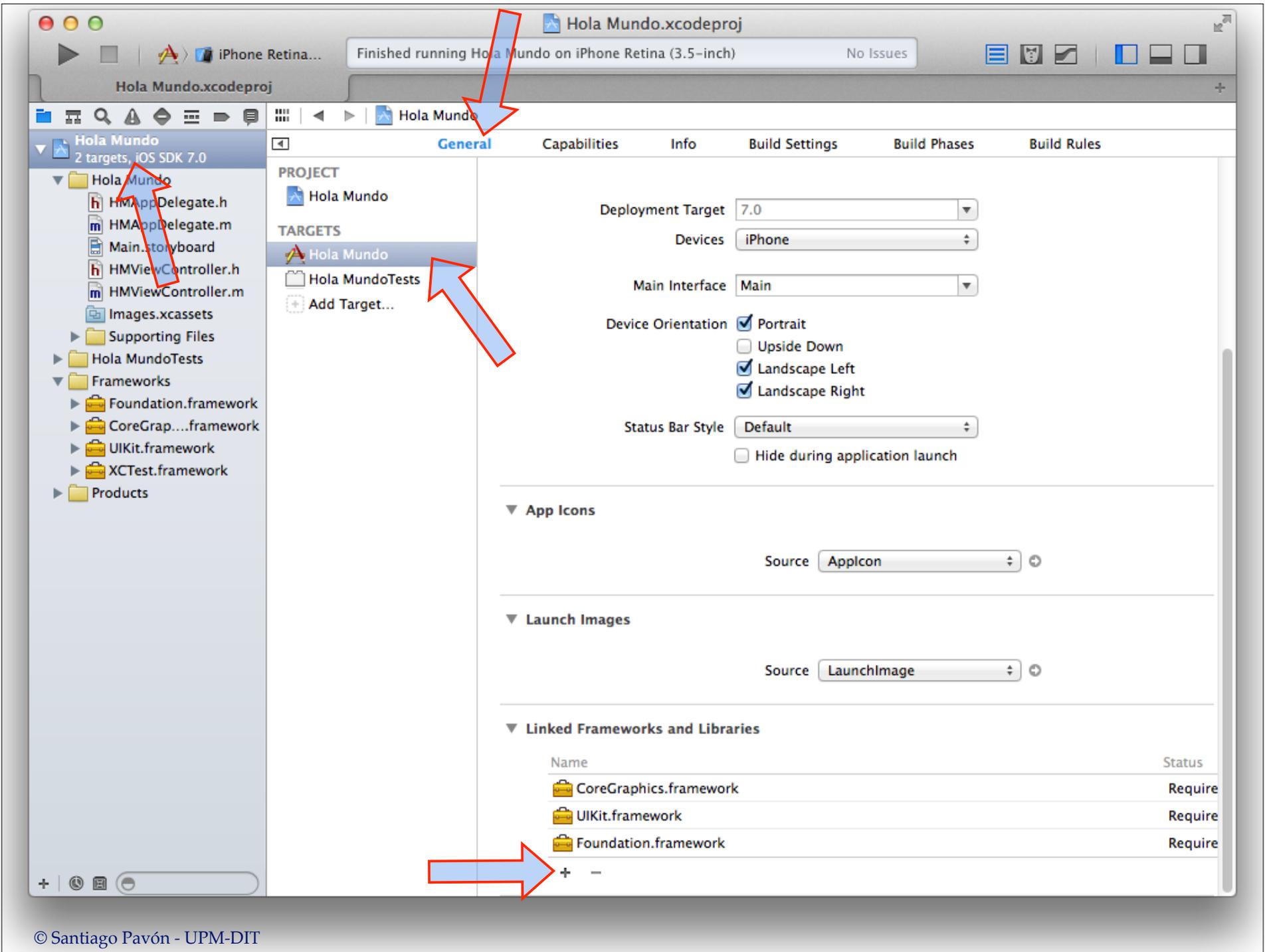
Ver el mundo

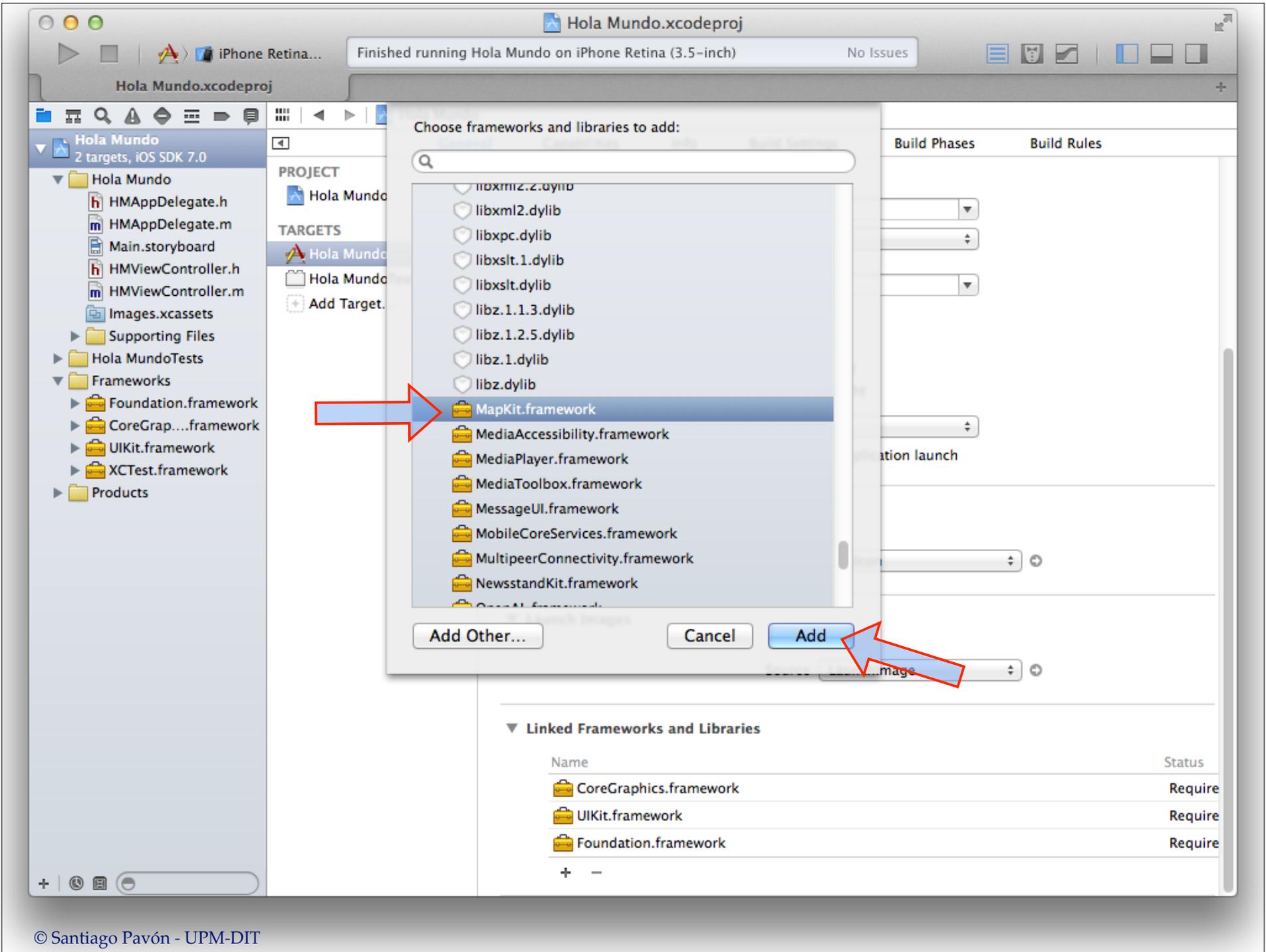
Si la aplicación se llama **Hola Mundo**, entonces quiero ver el mundo.

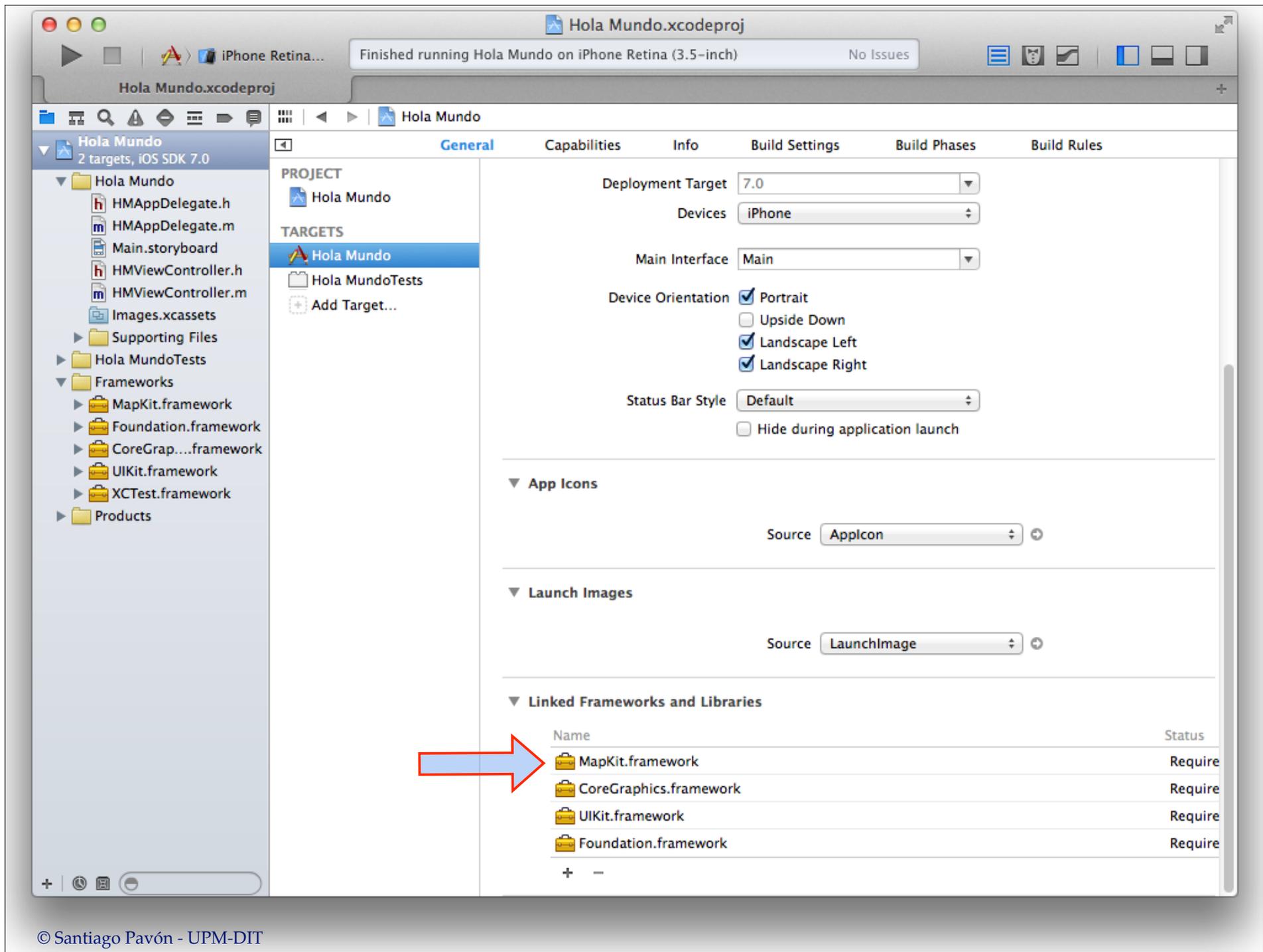
- Desde la librería de objetos añadimos un `MKMapView` a la vista.
- Hay que importar el framework `MapKit`











Teleco

- Al pulsar el botón **Mundo** quiero ver donde está Teleco en el mapa.
- Creamos un outlet al mapa y lo conectamos.
 - Sabemos hacerlo.
 - Es igual que con la UILabel.
- Desde **mundopressed** ponemos las coordenadas del aula en el mapa.
- No olvidar incluir:

```
#import <MapKit/MapKit.h>
```



```
//
//  HMViewController.m
//  Hola Mundo
//
//  Created by Santiago Pavón on 06/02/14.
//  Copyright (c) 2014 UPM. All rights reserved.
//

#import "HMViewController.h"
#import <MapKit/MapKit.h>

@interface HMViewController ()
@property (weak, nonatomic) IBOutlet UILabel *msg;
@property (weak, nonatomic) IBOutlet MKMapView *map;
@end

@implementation HMViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (IBAction)holaPressed:(UIButton *)sender {
    self.msg.text = @"Hola";
}

- (IBAction)mundoPressed:(UIButton *)sender {
    self.msg.text = @"Mundo";
    MKCoordinateRegion reg = {{40.452445, -3.726162}, {0.002, 0.002}};
    [self.map setRegion:reg animated:YES];
}

- (IBAction)sliderMoved:(UISlider *)sender {
    self.msg.alpha = sender.value;
}

- (void)didReceiveMemoryWarning
```

Práctica 1

- Hacer la demo Hola Mundo con los siguientes cambios:
 - Al pulsar el botón Hola:
 - El slider debe ajustarse a 0.5
 - El mapa debe configurarse en modo híbrido.
 - Al pulsar el botón Mundo:
 - El mapa debe configurarse en modo satélite.
 - Añadir otro botón para mostrar las pirámides de Egipto.

