



POLITÉCNICA

ETSIT
UPM

dit
UPM

Desarrollo de Apps para iOS

Scroll View

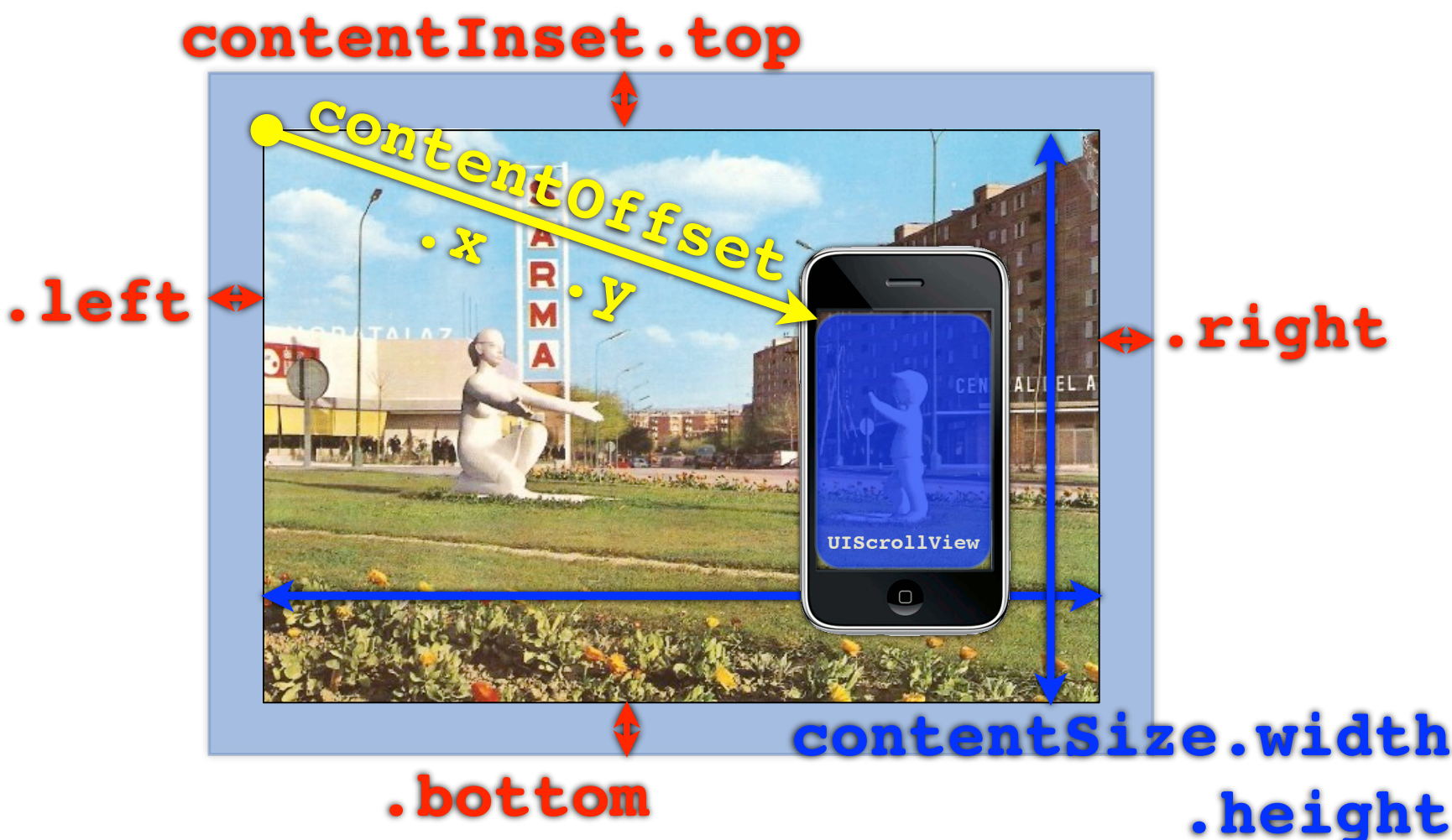
IWEB,LSWC 2013-2014

Santiago Pavón

ver: 2013.10.19

UIScrollView

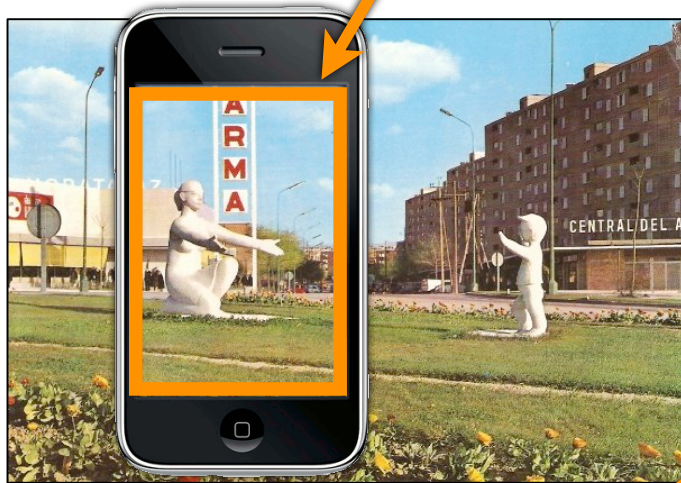
- Mostrar contenidos que no caben en la pantalla.
- Añadimos subviews que sobrepasan los límites de la UIScrollView contenedora.
- Podemos:
 - desplazar el contenido
 - hacer zoom





`scrollIndicatorInset`
`.right`
`.bottom`

CGRect b = scrollView.bounds



**CGRect r = [scrollView convertRect:b
toView:subview]**

Nota: **r** es mayor que **b** porque hay zoom

Propiedades de UIScrollView

`scrollView.contentSize`
`height, width`

`scrollView.contentInset`
`top, bottom, left, right`

`scrollView.contentOffset`
`x, y`

`scrollView.scrollIndicatorInset`
`top, bottom, left, right`

`etc . . .`

y las heredadas de las superclases

Crear ScrollView

- Crear Scroll View programáticamente:

```
CGRect frame = CGRectMake(0,0,300,300);  
UIScrollView* scrollview = [[UIScrollView alloc]  
                           initWithFrame:frame];
```

- Dos formas de crear un Scroll View en Xcode
 - Arrastrar un objeto UIScrollView desde la librería hasta el storyboard
 - Elegir una view en el storyboard y embeberla en un Scroll View
 - (Editor > Embed In > Scroll View)
 - En ambos casos crearemos un outlet que apunte al Scroll View para manejarlo programáticamente.

El contenido del Scroll View

- Añadir la jerarquía de subviews al Scroll View

```
UIImage* imagen = [UIImage imageNamed:@"sarma.jpg"];  
UIImageView* iv = [[UIImageView alloc]  
                  initWithImage:imagen];  
[scrollView addSubview:iv];
```

- También puede añadirse el contenido desde Xcode.
 - Poner el tamaño de la view gestionada a su valor real con el inspector.

Tamaño del contenido

- Siempre hay que indicar cual es el tamaño del contenido que está gestionando el Scroll View

- Tanto si se ha creado el Scroll View con Xcode o programáticamente.

```
scrollview.contentSize = imagen.size;
```

- Con **iOS 7**, si usamos **autolayout** no es necesario asignar un valor a **contentSize**.

- Se asigna automáticamente en función de las restricciones de las subviews añadidas al ScrollView.

UIScrollViewDelegate

- Los objetos scroll view informan al delegado:
 - cuando han realizado un desplazamiento.
 - cuando empiezan y terminan las aceleraciones.
 - etc.
- Le preguntan si:
 - deben volver al principio al llegar al final del desplazamiento.
 - sobre que view debe hacerse el zoom
 - etc.

UIScrollViewDelegate

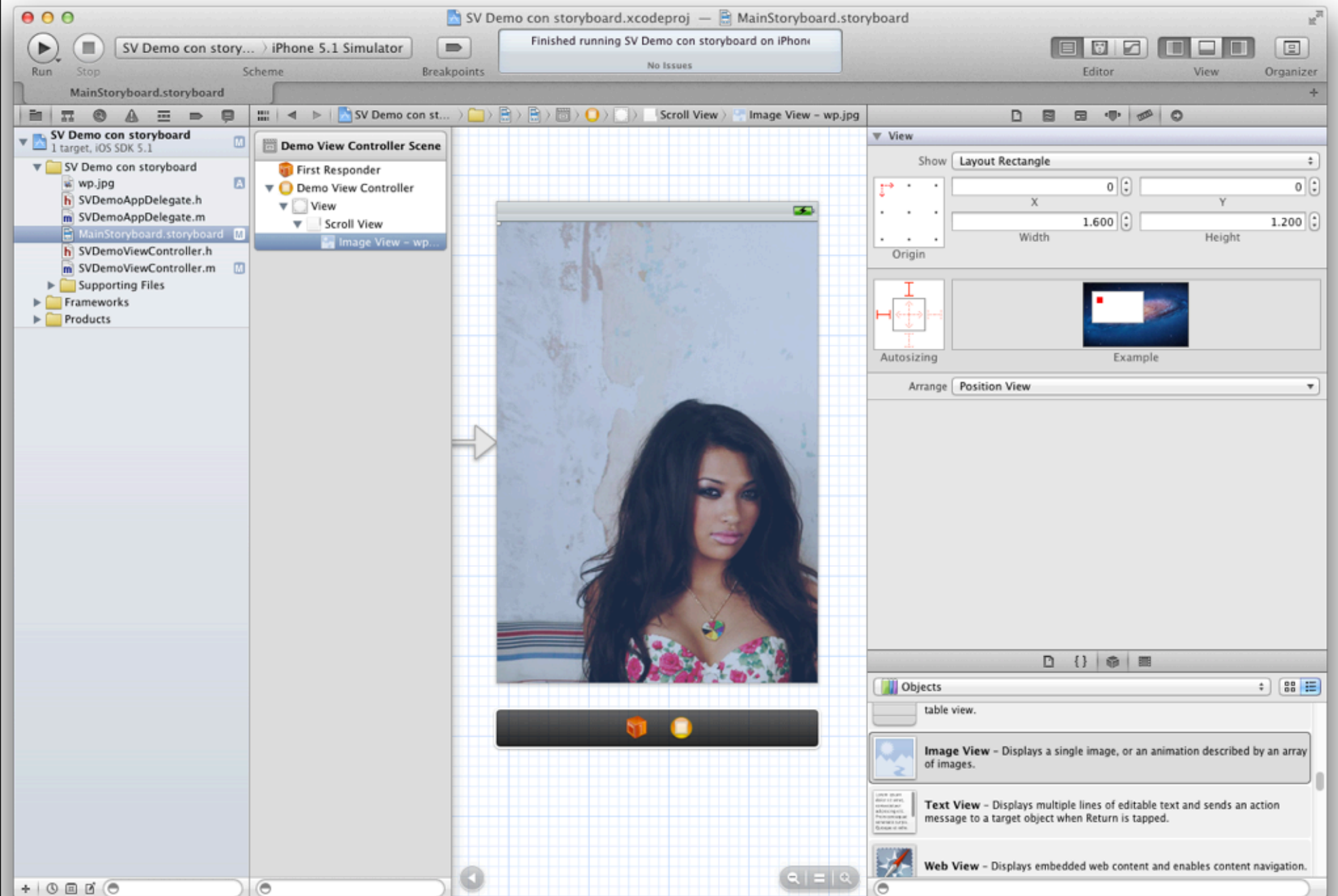
- Responding to Scrolling and Dragging
 - `scrollViewDidScroll:`
 - `scrollViewWillBeginDragging:`
 - `scrollViewDidEndDragging:willDecelerate:`
 - `scrollViewShouldScrollToTop:`
 - `scrollViewDidScrollToTop:`
 - `scrollViewWillBeginDecelerating:`
 - `scrollViewDidEndDecelerating:`
- Managing Zooming
 - `viewForZoomingInScrollView:`
 - `scrollViewDidEndZooming:withView:atScale:`
- Responding to Scrolling Animations
 - `scrollViewDidEndScrollingAnimation:`

Zoom

- Se basa en aplicar una transformación affine a la subview.
 - Modifica su propiedad **transform**.
- Propiedades:
 - `scrollview.maximumZoomScale`
 - `scrollview.minimumZoomScale`
 - Si no se les asigna un valor, el zoom no funcionará.
- Métodos:
 - `-(void)setZoomScale:(float) animated:(BOOL);`
 - `-(void)zoomToRect:(CGRect) animated:(BOOL);`
- Obligatorio asignar el delegado e implementar en él el siguiente método para que el zoom funcione:
 - `-viewForZoomingInScrollView:(UIScrollView*) sender`

Demostración - iOS 6

- Crear una aplicación basada en Simple View Application
- En el Storyboard:
 - Añadir un UIScrollView a la view.
 - Crear un outlet que la apunte.
- Sobrecribir viewDidLoad:
 - añadir una UIImageView al Scroll View con el contenido que se quiera.
 - configurar tamaño del contenido (contentSize).
- Programar el zoom:
 - Enganchar el delegado del Scroll View al View Controller.
 - El VC implementará el método viewForZoomingInScrollView:
 - devuelve la UIImageView.
 - UIImageView será una propiedad.
 - En viewDidLoad configurar el zoom máximo y mínimo.



The image shows the Xcode IDE interface. On the left, the storyboard is visible with a scene named "Demo View Controller Scene" containing a "View" with a "Scroll View" and an "Image View - wp.jpg". Two callout boxes labeled "Outlet" point to the "Image View" and "Scroll View" respectively. Below the storyboard, two panels show the "Referencing Outlets" for "Image View - wp.jpg" and "Scroll View". The "Image View" panel shows an outlet for "imageView" connected to "Demo View Controller". The "Scroll View" panel shows outlets for "delegate" and "scrollView", both connected to "Demo View Controller". A callout box labeled "Delegado y outlet" points to the "delegate" outlet in the "Scroll View" panel.

On the right, the code editor shows the implementation of `SVDemoViewController`. A callout box labeled "Outlets" points to the `@property` declarations for `UIScrollView` and `UIImageView`. A callout box labeled "contentSize" points to the `self.scrollView.contentSize = self.imageView.image.size;` line in the `viewDidLoad` method. A callout box labeled "Zoom máximo y mínimo" points to the `self.scrollView.minimumZoomScale` and `self.scrollView.maximumZoomScale` lines. A callout box labeled "View de la que hacemos zoom" points to the `viewForZoomingInScrollView` method.

```
// SVDemoViewController.m
// SV Demo con storyboard
//
// Created by Santiago Pavón Gómez on 18/06/12.
// Copyright (c) 2012 UPM - ETSIT. All rights reserved.

#import "SVDemoViewController.h"

@interface SVDemoViewController () <UIScrollViewDelegate>
@property (weak, nonatomic) IBOutlet UIScrollView *scrollView;
@property (weak, nonatomic) IBOutlet UIImageView *imageView;
@end

@implementation SVDemoViewController

@synthesize scrollView = _scrollView;
@synthesize imageView = _imageView;

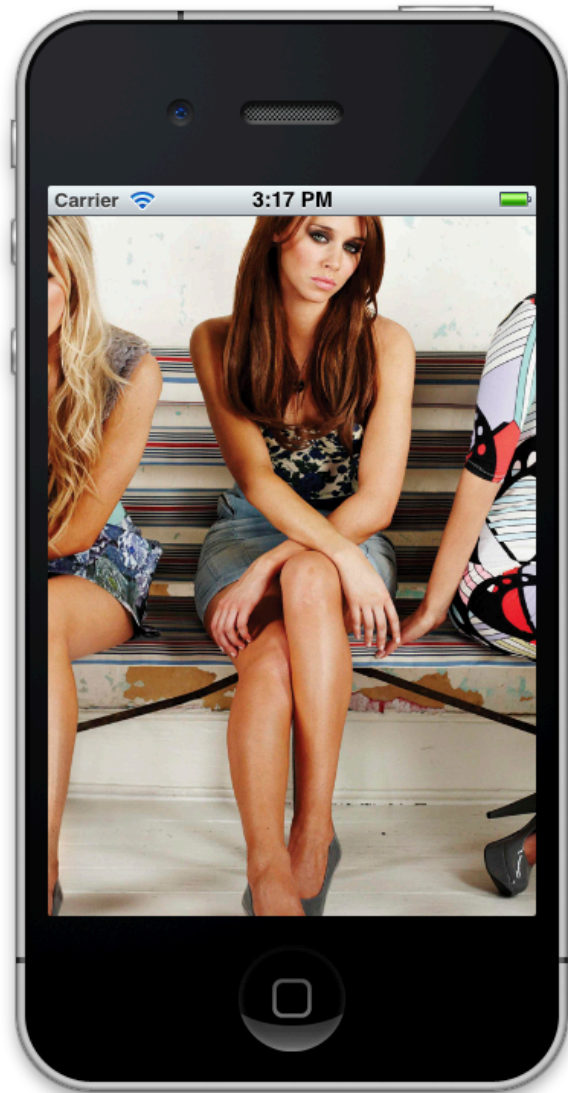
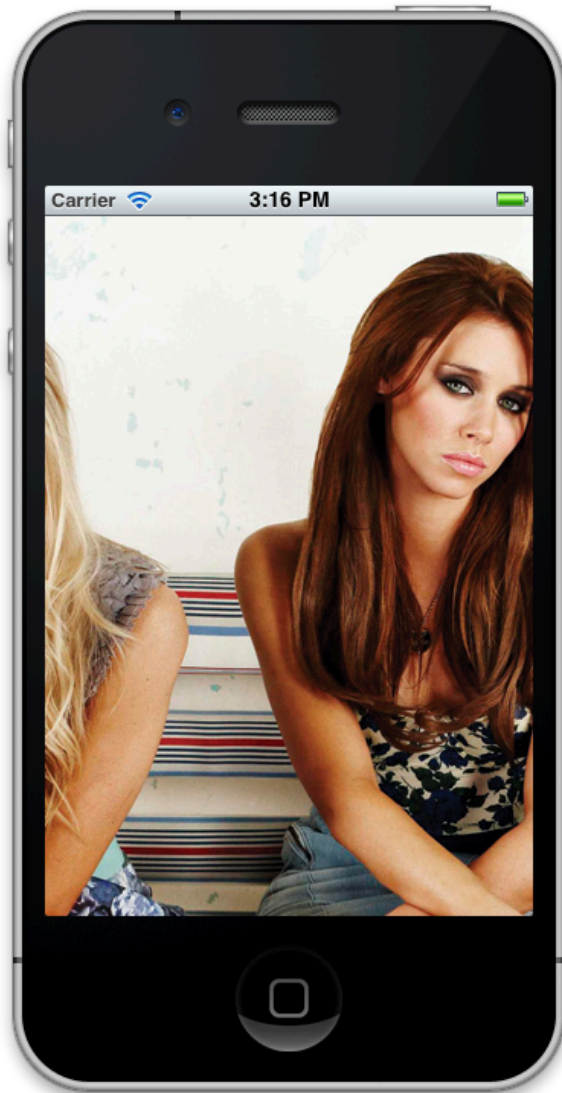
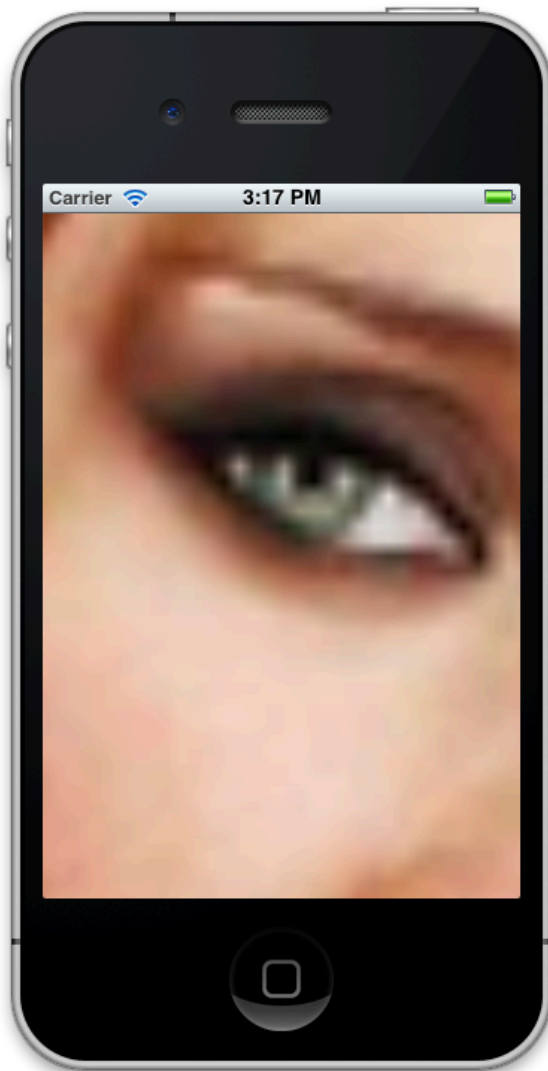
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    self.scrollView.contentSize = self.imageView.image.size;
    self.scrollView.minimumZoomScale = self.scrollView.bounds.size.width /
        self.imageView.bounds.size.width;
    self.scrollView.maximumZoomScale = 10;
}

- (void)viewDidUnload
{
    [self setScrollView:nil];
    [self setImageView:nil];
    [super viewDidUnload];
    // Release any retained subviews of the main
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)
interfaceOrientation
{
    return (interfaceOrientation != UIInterfaceOrientationPortraitUpsideDown);
}

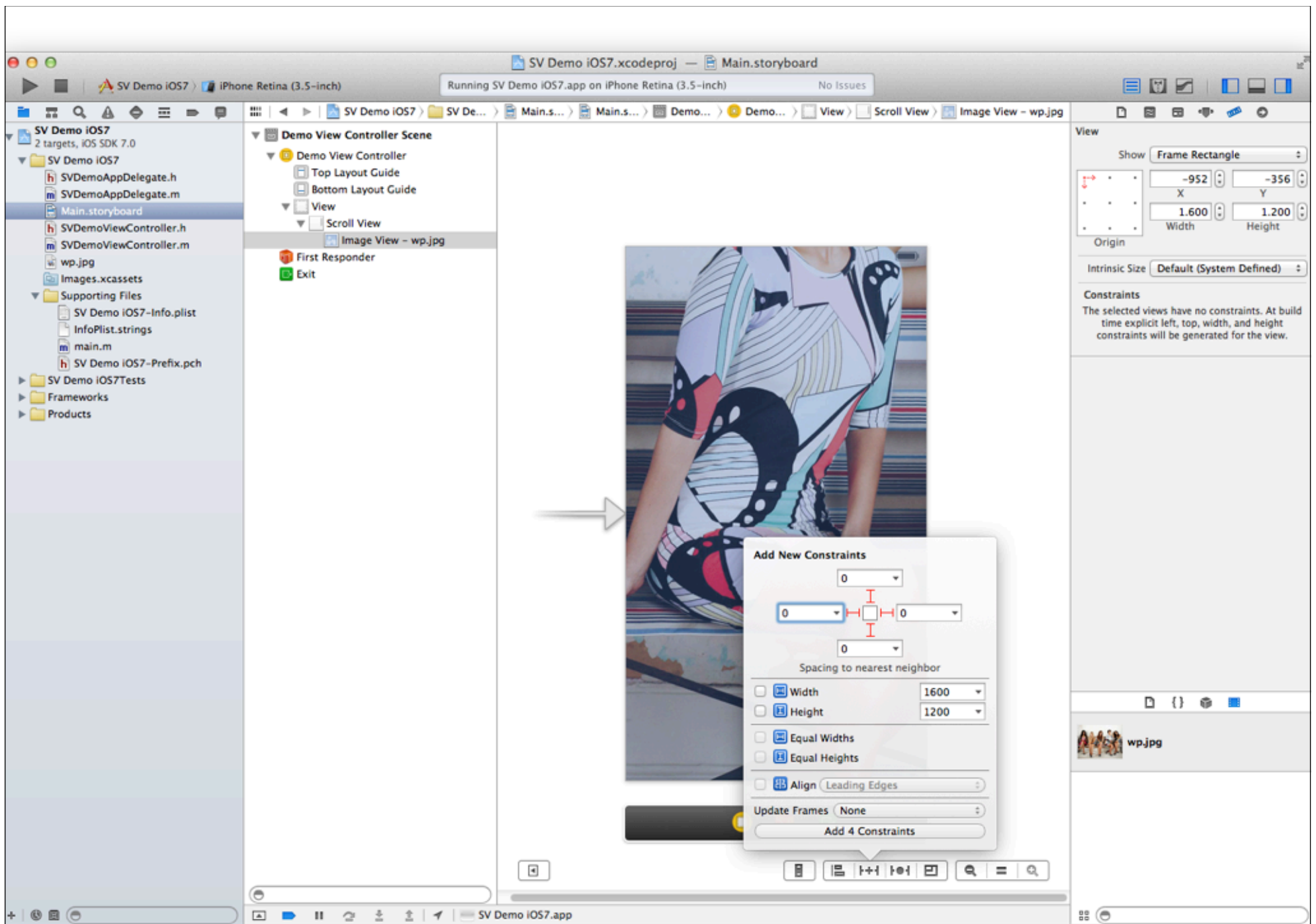
- (UIView *)viewForZoomingInScrollView:(UIScrollView *)scrollView {
    return self.imageView;
}

@end
```



Demostración - iOS 7

- Crear una aplicación basada en Simple View Application
- En el Storyboard:
 - Usar autolayout.
 - Añadir un UIScrollView a la view.
 - Añadir una imagen al UIScrollView.
 - Se crea una UIImageView automáticamente.
 - Seleccionar la UIImageView y crear 4 restricciones:
 - Spacing To nearest neighbor = 0, 0, 0 y 0
 - Así no es necesario poner el valor de `contentSize`.
 - Probarlo en el simulador.
- Programar el zoom:
 - Crear un outlet que apunte al Scroll View.
 - Crear un outlet que apunte a la Image View.
 - Enganchar el delegado del Scroll View al View Controller.
 - Engancharlo usando el storyboard o programáticamente.
 - El VC implementará el método `viewForZoomingInScrollView`:
 - devuelve el outlet UIImageView.
 - En `viewDidLoad` configurar el zoom máximo y mínimo.



```
SV Demo iOS7.xcodeproj — SVDemoViewController.m
Running SV Demo iOS7 on iPhone Retina (3.5-inch) No Issues
SV Demo iOS7 > SV Demo iOS7 > SVDemoViewController.m > No Selection
//
#import "SVDemoViewController.h"

@interface SVDemoViewController () <UIScrollViewDelegate>
@property (weak, nonatomic) IBOutlet UIScrollView *scrollView;
@property (weak, nonatomic) IBOutlet UIImageView *imageView;
@end

@implementation SVDemoViewController

- (void)viewDidLoad
{
    [super viewDidLoad];

    // Reduzco hasta ver ancho de toda la foto y amplio hasta por 10.
    self.scrollView.minimumZoomScale = self.scrollView.bounds.size.width /
                                        self.imageView.bounds.size.width;
    self.scrollView.maximumZoomScale = 10;

    self.scrollView.delegate = self;
}

#pragma mark - Scroll View Delegate

- (UIView*) viewForZoomingInScrollView:(UIScrollView *)scrollView
{
    return self.imageView;
}

@end
```

