



POLITÉCNICA

ETSIT
UPM

dit
UPM

Desarrollo de Apps para iOS

Scroll View

IWEB 2015-2016
Santiago Pavón

ver: 2015.10.08

UIScrollView

- Mostrar una view (o varias views) con un contenido de gran tamaño que no cabe en la pantalla.
- La añadimos como una subview de un ScrollView.
- Podemos:
 - desplazarnos por el contenido.
 - hacer zoom.



Propiedades de UIScrollView

`scrollView.contentSize`
height, width

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

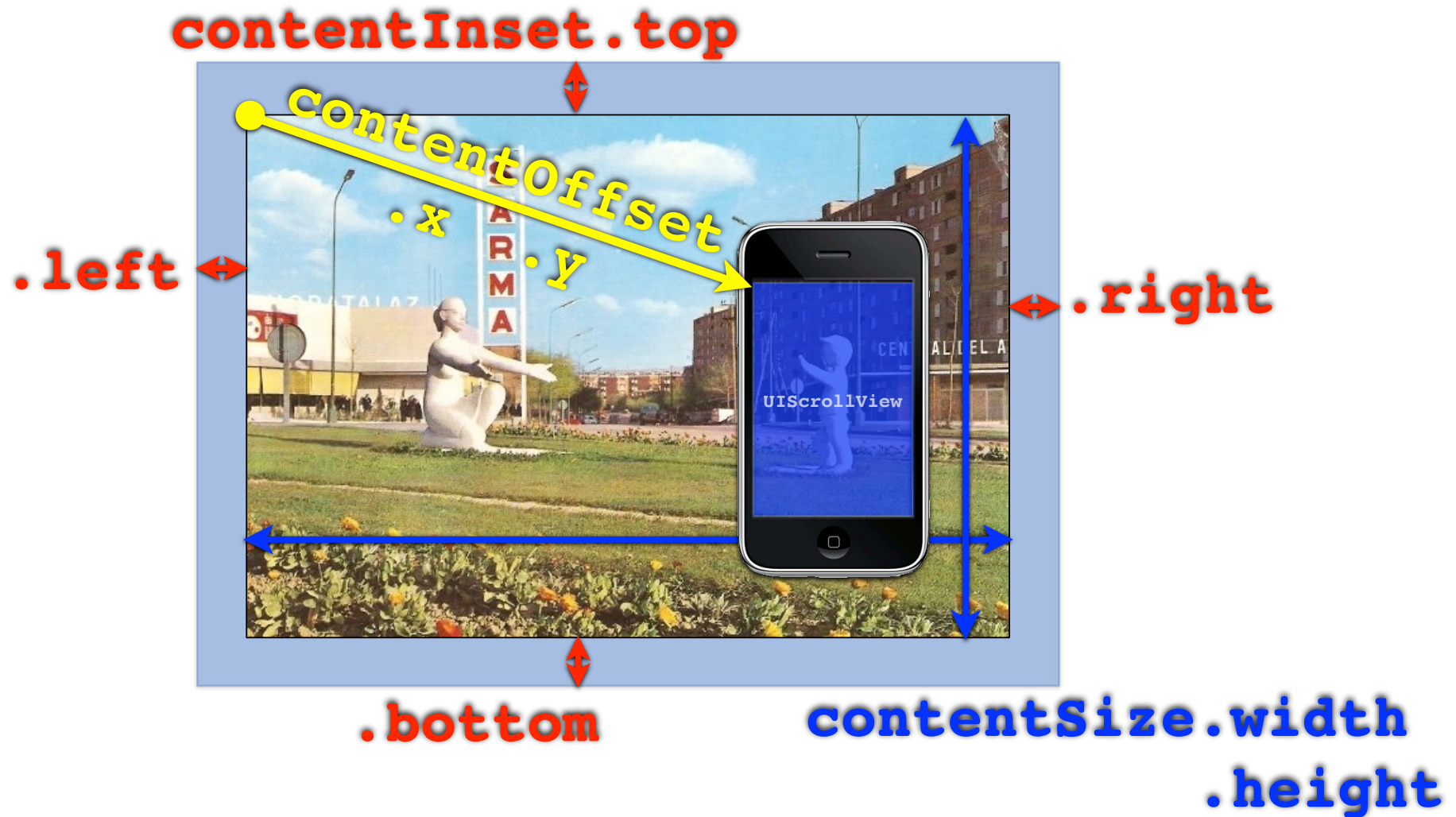
`scrollView.contentOffset`
x, y

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

etc . . .

y las heredadas de las superclases

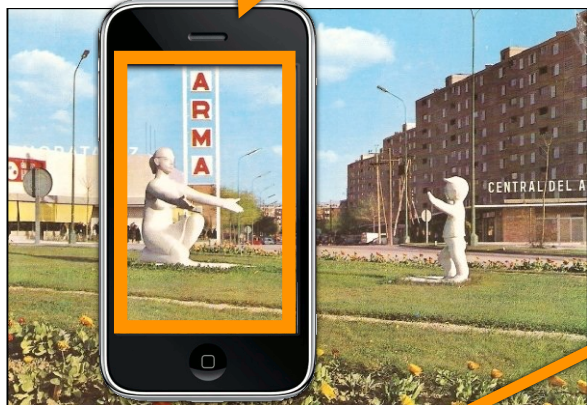
- Para que el UIScrollView funcione, debe asignarse el tamaño del contenido a la propiedad `contentSize` del Scroll View.





`scrollIndicatorInset`
`.right`
`.bottom`

```
let b: CGRect = scrollView.bounds
```



```
let r: CGRect = scrollView.convertRect(b, toView: subview)
```

Si hacemos zoom, el bounds del ScrollView será de distinto tamaño que el rectángulo visto de la subview.

r es mayor que **b** si hay zoom.

Crear ScrollView Programáticamente

- Creamos el objeto ScrollView:

```
let rect = CGRectMake(0, 0, 300, 300)
```

```
let scrollView = UIScrollView(frame: rect)
```

- Añadir la jerarquía de subviews al Scroll View

```
let img = UIImage(named: "sarma.jpg")
```

```
let imgview = UIImageView(image: img!)
```

```
scrollView.addSubview(imgview)
```

- Indicar cuál es el tamaño del contenido:

```
scrollView.contentSize = img!.size
```

Crear ScrollView con IB

- Dos formas de crear un ScrollView con IB:
 - Arrastrar un objeto UIScrollView desde la librería hasta el storyboard.
 - Y añadir las subviews al UIScrollView.
 - Elegir una view ya existente en el storyboard y embeberla en un ScrollView.

menú Editor > Embed In > Scroll View

- Poner el tamaño del contenido de la subview gestionada:
 - Poner su valor programáticamente (*crear un outlet para ello*)
 - o usar autolayout para indicar su posición y tamaño.

Tamaño del contenido

MUY IMPORTANTE

- Siempre hay que indicar cuál es el tamaño del contenido que está gestionando el ScrollView.

```
scrollView.contentSize = imagen.size;
```

- Tanto si se ha creado el ScrollView con Interface Builder o programáticamente.

- Pero si usamos autolayout para colocar las subviews que gestiona el ScrollView, no es necesario asignar un valor a `contentSize`.

- Se calcula automáticamente en función de las restricciones de añadidas a las subviews.

UIScrollViewDelegate

- Los objetos UIScrollView 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 subview debe hacerse el zoom.
 - etc.

UIScrollViewDelegate

- Responding to Scrolling and Dragging

```
scrollViewDidScroll(:)
scrollViewWillBeginDragging(:)
scrollViewDidEndDragging(,willDecelerate:)
scrollViewShouldScrollToTop(:) -> Bool
scrollViewDidScrollToTop(:)
scrollViewWillBeginDecelerating(:)
scrollViewDidEndDecelerating(:)
```

- Managing Zooming

```
viewForZoomingInScrollView(:) -> UIView?
scrollViewWillBeginZooming(,withView:)
scrollViewDidEndZooming(,withView:,atScale:)
scrollViewDidZoom(:)
```

- Responding to Scrolling Animations

```
scrollViewDidEndScrollingAnimation(:)
```


Zoom

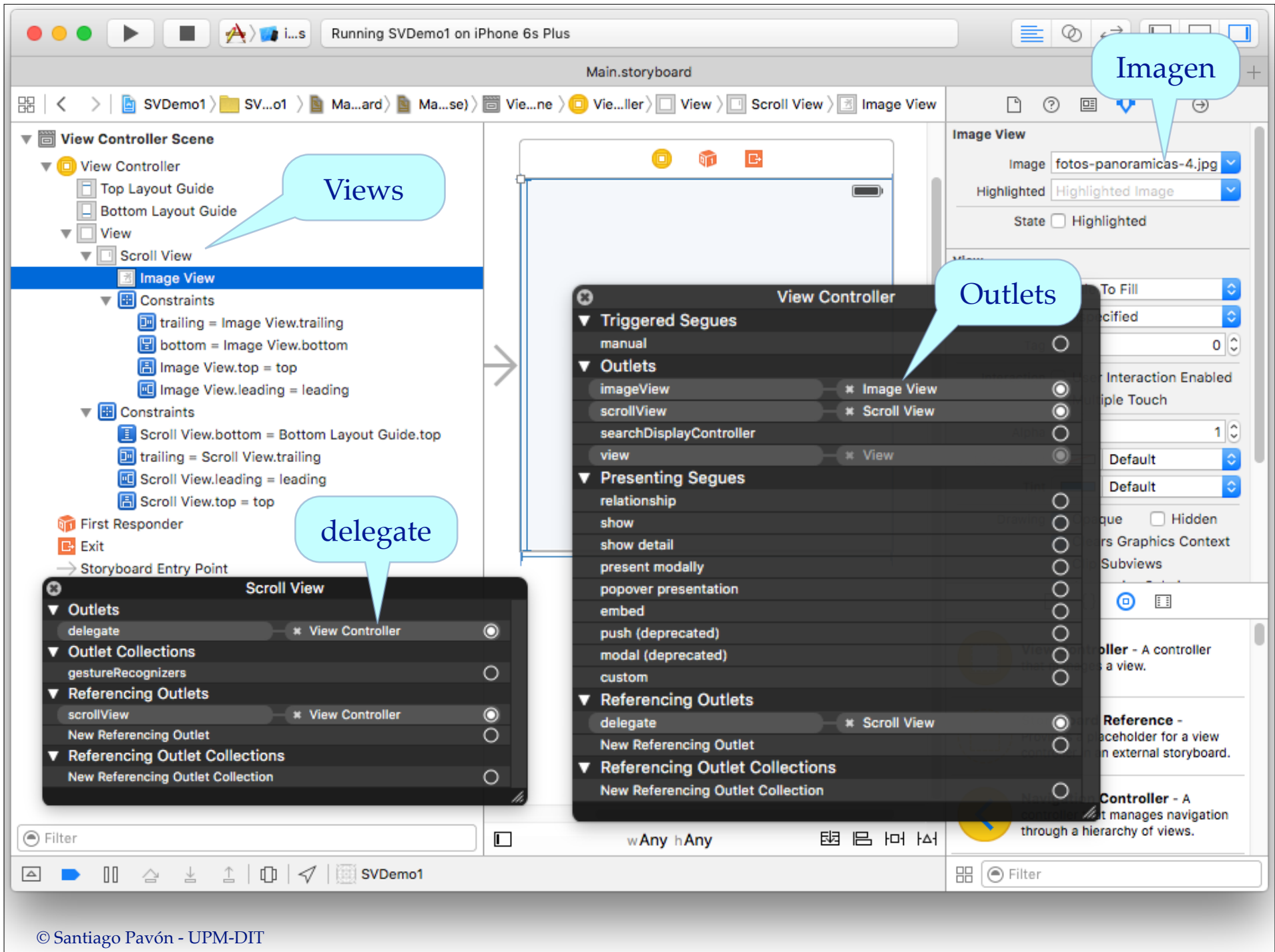
- Se basa en aplicar una transformación affine a la subview.
 - Modifica su propiedad **transform**.
- Propiedades:
 - `maximumZoomScale`
 - `minimumZoomScale`
 - **IMPORTANTE**: Si no se les asigna un valor, el zoom no funcionará.
- Métodos:
 - `setZoomScale(:, animated:)`
 - `zoomToRect(:, animated:)`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

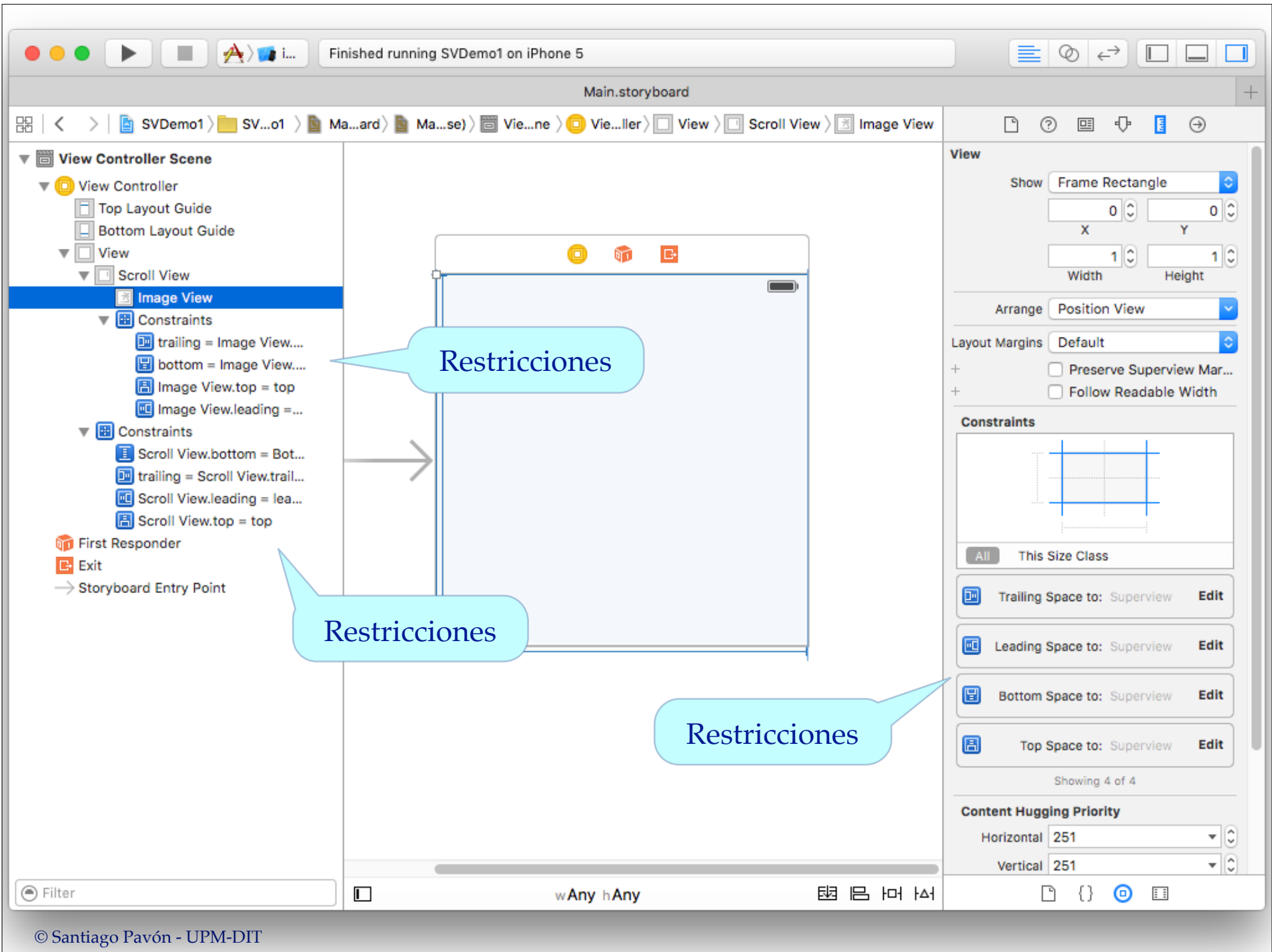
Demo

Pasos

- Crear una aplicación basada en Simple View Application.
- Editar Main.storyboard (con IB):
 - Añadir una UIScrollView a la view.
 - Añadir restricciones para que ocupe toda la pantalla (su superview).
 - Crear un outlet que apunte a la UIScrollView.
 - Añadir una UIImageView como subview de la UIScrollView.
 - Usar una imagen muy grande para visualizarla en la UIImageView.
 - Establecer el layout de la UIImageView:
 - Crear cuatro restricciones para indicar que la separación arriba, abajo y a los lados con su superview (la ScrollView) es cero.
 - Nota: Al usar autolayout para configurar el tamaño de la UIImageView, no es necesario programar que `scrollView.contentSize` sea igual al tamaño de la imagen.

- Para programar el zoom:
 - Hacer que ViewController sea conforme con UIScrollViewDelegate.
 - Hacer que ViewController sea el delegado del ScrollView.
 - Puede hacerse de dos formas:
 - Añadiendo en viewDidLoad la sentencia
`scrollView.delegate = self.`
 - Con IB enganchando el delegado del ScrollView con el VC.
 - Implementar el método `viewForZoomingInScrollView:` en ViewController.
 - Este método devuelve la UIImageView.
 - Necesitamos para ello crear un outlet que apunte a UIImageView.
 - Configurar el zoom máximo y mínimo en `viewWillAppear.`
 - También se puede hacer con el inspector de atributos en el IB.





Finished running SVDemo1 on iPhone 6s Plus

ViewController.swift

```
import UIKit

class ViewController: UIViewController, UIScrollViewDelegate {

    @IBOutlet weak var scrollView: UIScrollView!
    @IBOutlet weak var imageView: UIImageView!

    override func viewDidLoad() {
        super.viewDidLoad()
    }

    override func viewWillAppear(animated: Bool) {
        super.viewWillAppear(animated)

        scrollView.minimumZoomScale = scrollView.bounds.size.width /
            imageView.bounds.size.width
        scrollView.maximumZoomScale = 10
    }

    func viewForZoomingInScrollView(scrollView: UIScrollView) -> UIView? {
        return imageView
    }

    override func viewWillTransitionToSize(size: CGSize, withTransitionCoordinator
        coordinator: UIViewControllerTransitionCoordinator) {
        scrollView.minimumZoomScale = size.width / imageView.bounds.size.width
        scrollView.setZoomScale(scrollView.zoomScale, animated: true)
    }
}
```

Protocolo delegate

Outlets

Zoom Mínimo y Máximo

View de la que hacemos zoom

© Santiago Pavón - UPM-DIT



