ETSIT UPM

dit UPM

Desarrollo de Apps para iOS

Scroll View

IWEB 2016-2017
Santiago Pavón

ver: 2016.10.03

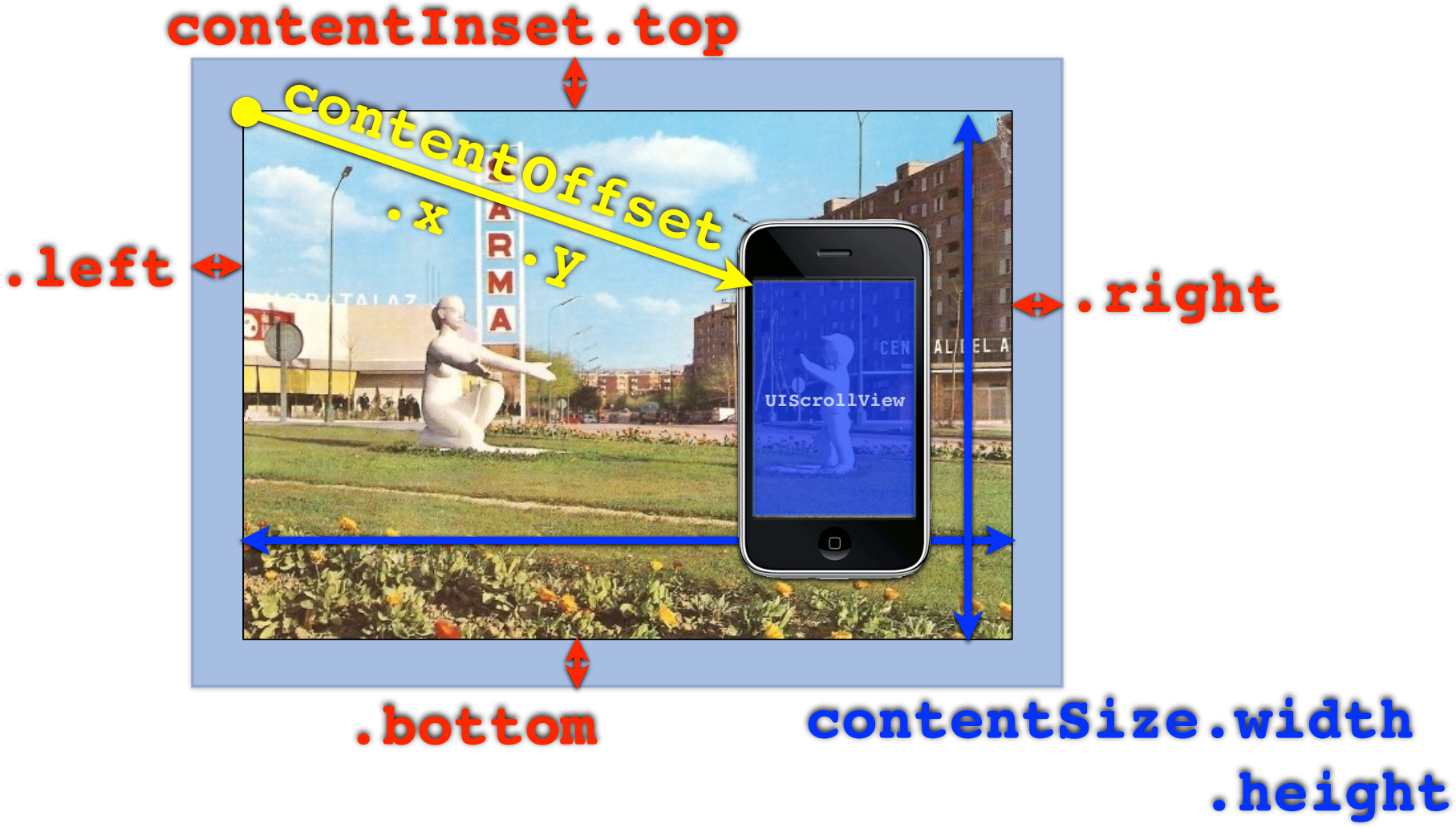
UIScrollView

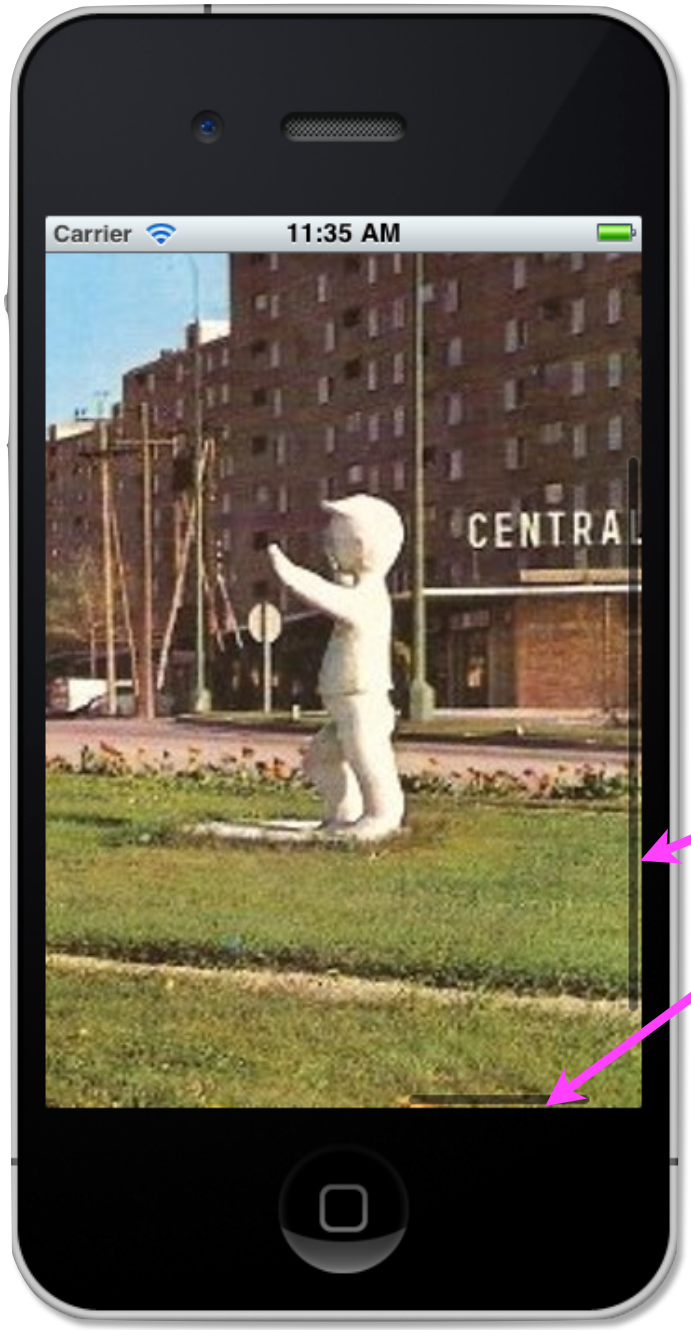
- Mostrar una view (o varias views) con un contenido que tiene un tamaño que no cabe en la pantalla.
- La añadimos como una subview de un ScrollView.
- Podemos:
  - desplazarnos por el contenido.
  - hacer zoom.

Propiedades de UIScrollView

```
var contentSize: CGSize
      height, width

var contentInset: UIEdgeInsets
      top, bottom, left, right

var contentOffset: CGPoint
      x, y

var scrollIndicatorInset: UIEdgeInsets
      top, bottom, left, right

etc . . .

y las heredadas de las superclases
```

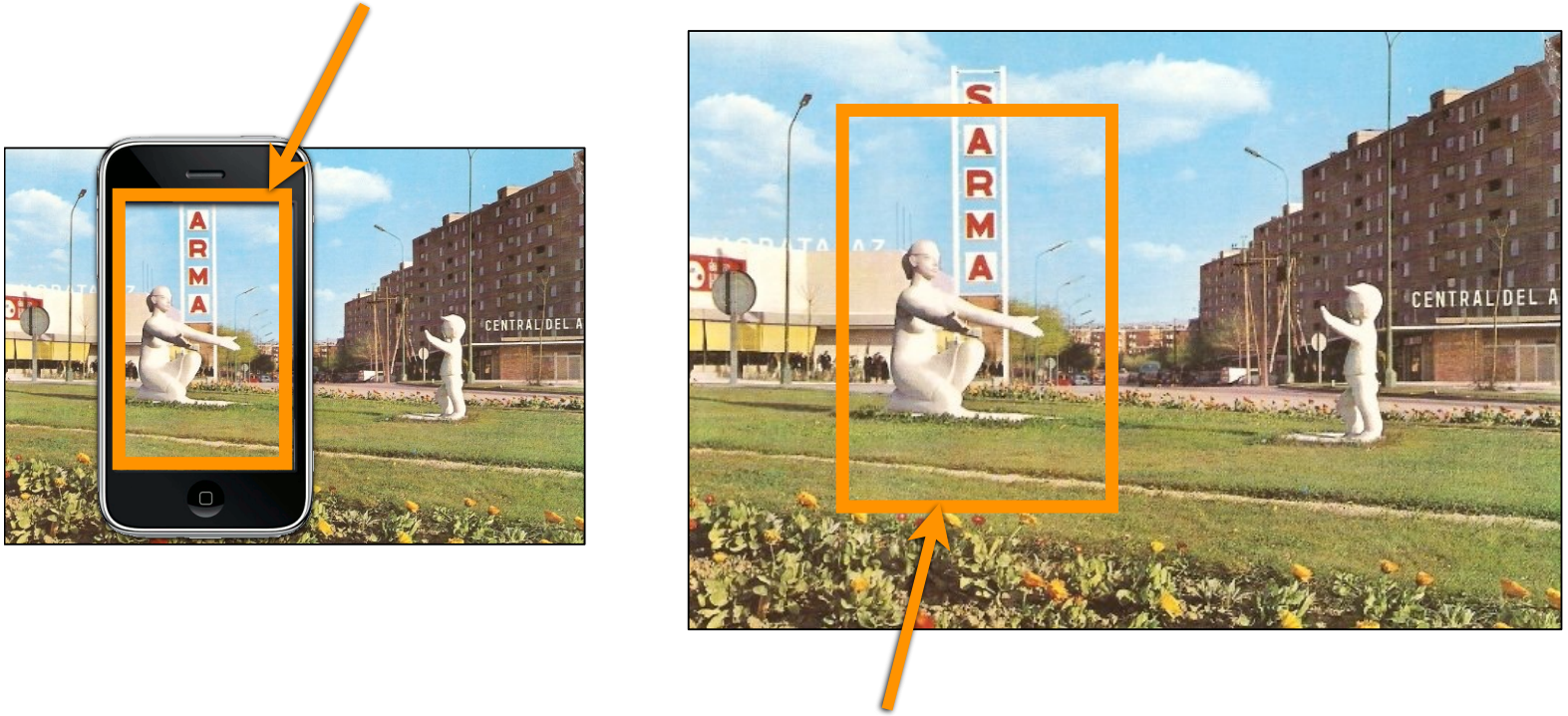
- Para que el UIScrollView funcione, debe asignarse el tamaño del contenido a la propiedad **contentSize** del Scroll View.






```
scrollIndicatorInset
    .right
    .bottom
```

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




```
let r: CGRect = scrollview.convert(b, to: subview)
```

- **b** es el rectángulo con el tamaño del scrollview.
- **r** es el rectángulo de la zona que estamos viendo de la subview.
  - Si hacemos zoom, esos rectángulos son de distinto tamaño: Con zoom **r** es mayor que **b**.
  - El método convert de la UIViews nos permite hacer conversiones entre los sistemas de coordenadas.

Crear ScrollView Programáticamente

- Creamos un ScrollView y lo añadimos a view ocupando todo el espacio:

```
let rect = view.bounds
let scrollview = UIScrollView(frame: rect)
view.addSubview(scrollview)
```

- Añadir las 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 programaticamente *(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.

Más Propiedades y Métodos

- Existen muchas propiedades y métodos en la clase UIScrollView.
  - Propiedades que indican si estamos desplazando el contenido, si se está moviendo,…

    `isDragging`, `isTracking`, `isDecelerating`, …
  - Métodos para posicionar el contenido en una posición.

```
func scrollRectToVisible(CGRect,
                         animated: Bool)
```

  - etc…

UIScrollViewDelegate

- Los objetos ScrollView 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

```
func scrollViewDidScroll(UIScrollView)
func scrollViewWillBeginDragging(UIScrollView)
func scrollViewWillEndDragging(UIScrollView, withVelocity: CGPoint,
                targetContentOffset: UnsafeMutablePointer<CGPoint>)
func scrollViewDidEndDragging(UIScrollView, willDecelerate: Bool)
func scrollViewShouldScrollToTop(UIScrollView) -> Bool
func scrollViewDidScrollToTop(UIScrollView)
func scrollViewWillBeginDecelerating(UIScrollView)
func scrollViewDidEndDecelerating(UIScrollView)
```

- Managing Zooming

```
func viewForZooming(in: UIScrollView) -> UIView?
func scrollViewWillBeginZooming(UIScrollView, with: UIView?)
func scrollViewDidEndZooming(UIScrollView, with: UIView?,
                             atScale: CGFloat)
func scrollViewDidZoom(UIScrollView)
```

- Responding to Scrolling Animations

```
func scrollViewDidEndScrollingAnimation(UIScrollView)
```

Zoom

- Se basa en aplicar una transformación affine a la subview.
  - Modifica su propiedad **transform**.
- Propiedades:

```
var maximumZoomScale: CGFloat
var minimumZoomScale: CGFloat
```

  - **IMPORTANTE**: Si no se les asigna un valor, el zoom no funcionará.
- Métodos:

```
func setZoomScale(CGFloat, animated: Bool)
func zoom(to: CGRect, animated: Bool)
. . .
```

- Obligatorio asignar el delegado e implementar en él el siguiente método para que el zoom funcione:

```
func viewForZooming(in: UIScrollView) -> 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.
    - 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 `viewForZooming(inscrollView:) -> UIView?` en ViewController.
    - Este método debe devolver la UIImageView.
    - Hay que crear también un outlet que apunte a UIImageView.
  - Configurar el zoom máximo y mínimo en viewDidAppear.
    - También se puede hacer con el inspector de atributos en el IB.
  - Sobrescribimos el método `viewWillTransition(to size:, with coordinator:)` para reajustar el zoom mínimo y actual cuando cambia el tamaño (por una rotación).