ETSIT UPM

dit UPM

Desarrollo de Apps para iOS
Map Kit

IWEB,LSWC 2013-2014
Santiago Pavón

ver: 2012.09.04 p1

Características

- Proporciona una view que muestra mapas.
  - proporcionados por Google Earth.
- Tipos de visualización: mapa, satélite e híbrida.
- El usuario puede cambiar la región mostrada y hacer zoom con gestos.
- Permite mostrar la posición del terminal.
- Permite anotar los mapas.
- Busca la dirección de una posición.

Algunos tipos

```
typedef double CLLocationDegrees;

typedef struct {
      CLLocationDegrees latitude;
      CLLocationDegrees longitude;
} CLLocationCoordinate2D;

typedef struct {
    CLLocationDegrees latitudeDelta;
    CLLocationDegrees longitudeDelta;
} MKCoordinateSpan;

typedef struct {
      CLLocationCoordinate2D center;
      MKCoordinateSpan span;
} MKCoordinateRegion;
```

- Medidas sobre la proyección 2D del planeta

```
typedef struct {
    double x;
    double y;
} MKMapPoint;

typedef struct {
    double width;
    double height;
} MKMapSize;

typedef struct {
    MKMapPoint origin;
    MKMapSize size;
} MKMapRect;
```

- Existen muchas funciones, constantes y tipos C en **`MapKit.h`** para trabajar con puntos, areas, etc.

```
MKMapPointForCoordinate
MKMapPointMake
MKMapSizeEqualToSize
MKMapSizeMake
MKMapRectIntersection
MKMapRectMake
MKCoordinateForMapPoint
MKCoordinateRegionForMapRect
MKMetersBetweenMapPoints
MKMapRectRemainder
MKRoadWidthAtZoomScale
MKMapSize MKMapSizeWorld;
MKMapRect MKMapRectWorld;
. . .
```

MKMapView

- Un objeto **`MKMapView`** en una view donde se muestra un mapa.
  - Incluir el framework MapKit en el proyecto.
  - Importar `<MapKit/MapKit.h>`
- Región mostrada en el mapa:

```
@property MKCoordinateRegion region
@property CLLocationCoordinate2D centerCoordinate
@property MKMapRect visibleMapRect

-(void)setRegion:(MKCoordinateRegion)region
        animated:(BOOL)animated
-(void)setCenterCoordinate:(CLLocationCoordinate2D)coordinate
                  animated:(BOOL)animated
-(void)setVisibleMapRect:(MKMapRect)mapRect animated:(BOOL)animated
-(MKCoordinateRegion)regionThatFits:(MKCoordinateRegion)region
. . .
```

MKMapView

- Tipo de mapa:

```
@property MKMapType mapType
```

  - Valores:

```
MKMapTypeStandard
MKMapTypeSatellite
MKMapTypeHybrid
```

- Control de las acciones permitidas al usuario:

```
@property (getter=isZoomEnabled)   BOOL zoomEnabled
@property (getter=isScrollEnabled) BOOL scrollEnabled
```

MKMapView

- Posición del terminal:

```
@property BOOL showsUserLocation
@property (readonly,
getter=isUserLocationVisible)
          BOOL userLocationVisible
@property (readonly) MKUserLocation *userLocation
```

  - `MKUserLocation` es conforme a `MKAnnotation` y lo usa el mapa para guardar la posición del terminal.
    - El usuario no creará un objeto de este tipo. Lo puede obtener preguntándole al mapa.
  - Se muestra en el mapa con un punto azul.

MKMapView

- Conversiones:

```
-(CGPoint)convertCoordinate:(CLLocationCoordinate2D)coord
                toPointToView:(UIView*)view;

-(CLLocationCoordinate2D)convertPoint:(CGPoint)point
                toCoordinateFromView:(UIView*)view;

-(CGRect)convertRegion:(MKCoordinateRegion)region
              toRectToView:(UIView*)view;

-(MKCoordinateRegion)convertRect:(CGRect)rect
                toRegionFromView:(UIView*)view;
```

MKMapViewDelegate

- `MKMapView` tiene un delegado:

```
@property id<MKMapViewDelegate> delegate
```

  - El mapa informa al delegado sobre la carga de datos en el mapa, cambios en la región visualizada, gestiona varias tareas de las anotaciones, . . .

- La región mostrada cambia:
  - `mapView:regionWillChangeAnimated:`
  - `mapView:regionDidChangeAnimated:`
- Cargando los mapas de Google Earth:
  - `mapViewWillStartLoadingMap:`
  - `mapViewDidFinishLoadingMap:`
  - `mapViewDidFailLoadingMap:withError:`
- Seguimiento de la posición del terminal:
  - `mapViewWillStartLocatingUser:`
  - `mapViewDidStopLocatingUser:`
  - `mapView:didUpdateUserLocation:`
  - `mapView:didFailToLocateUserWithError:`

- Gestión de las Annotation Views
  - `mapView:viewForAnnotation:`
  - `mapView:didAddAnnotationViews:`
  - `mapView:annotationView:calloutAccessoryControlTapped:`
- Arrastrar una Annotation View
  - `mapView:annotationView:didChangeDragState:`
    `fromOldState:`
- Selecionar Annotation Views
  - `mapView:didSelectAnnotationView:`
  - `mapView:didDeselectAnnotationView:`
- Gestión de Overlay Views
  - `mapView:viewForOverlay:`
  - `mapView:didAddOverlayViews:`

Anotaciones

Anotaciones

- En los mapas pueden mostrarse anotaciones.
- Tienen unas coordenadas, un título y un subtítulo.
- Se visualizan en el mapa usando objetos de la clase (o subclase) `MKAnnotationView`.
- Al tocarlas pueden mostrar un *callout* con su título, su subtítulo, accesorios a la izquierda y a la derecha, ...

MKMapView

- Anotaciones que se muestran en el mapa:

```
@property (readonly) NSArray *annotations;

-addAnnotation:
-addAnnotations:
-removeAnnotation:
-removeAnnotations:

-(MKAnnotationView*)viewForAnnotation:

-(MKAnnotationView*)dequeueReusableAnnotationViewWithIdentifier:

@property (copy) NSArray *selectedAnnotations;

-selectAnnotation:animated:
-deselectAnnotation:animated:
```

Necesarios porque **annotations** es readonly

MKAnnotation

- Es un protocolo

```
@property (readonly) CLLocationCoordinate2D coordinate;
@optional
-(NSString*) title;
-(NSString*) subtitle;
-(void)setCoordinate:(CLLocationCoordinate2D)newCoordinate;
```

- Las clases que se usan como anotaciones deben ser conformes a este protocolo.
  - Ya existen varias clases conformes a este protocolo:
    - `MKPlaceMark`, `MKUserLocation`, . . .
  - Las clases creadas por el desarrollador para ser mostradas en un mapa como anotaciones deben adoptar este protocolo.

MKAnnotationView

- Las anotaciones se muestran en el mapa usando objetos de la clase `MKAnnotationView`.
  - Esta clase deriva de `UIView`.
- Cuando el mapa va a mostrar una anotación, envía a su delegado el mensaje
  - `- (MKAnnotationView*) mapView:viewForAnnotation:`
  - Este método devuelve el objeto (`MKAnnotationView`) que debe usarse para representar la anotación.
  - Si devuelve `nil`, el mapa usa por defecto un objeto de la clase `MKPinAnnotationView`.
    - Su imagen es un pin.

MKAnnotationView - Propiedades

`@property (nonatomic,retain) id <MKAnnotation> **annotation**`
- Apunta a la anotación asociada sólo si la annotation view es visible.

`@property (nonatomic,retain) UIImage ***image**`
- La imagen usada para visualizar la annotation view.

`@property (nonatomic,getter=isEnabled) BOOL **enabled**`
- Indica si se atienden o no los eventos del usuario.
- Si es NO, no puede seleccionarse, no muestra su callout, ...

`@property (nonatomic) BOOL **canShowCallout**`
- Indica si se mostrará un callout cuando se toque sobre la annotation view.

`@property (nonatomic,getter=isSelected) BOOL **selected**`
- Indica si está seleccionada, en cuyo caso estará mostrando un callout.

`@property (nonatomic) CGPoint **centerOffset**`
- Para desplazar el punto donde se muestra la view.

`@property (nonatomic) CGPoint **calloutOffset**`
- Para desplazar el punto donde se muestra el callout.

MKAnnotationView - Propiedades

```
@property (nonatomic,retain) UIView *leftCalloutAccessoryView
@property (nonatomic,retain) UIView *rightCalloutAccessoryView
```

- View mostrada a la izquierda/derecha del callout.
- Si asigno un UIControl, al tocarlo se envía al delegado el mensaje mapView:annotationView:calloutAccessoryControlTapped:

```
@property (nonatomic,getter=isDraggable) BOOL draggable
```

- Indica si se permite al usuario desplazar interactivamente la posición de la annotation view.
- Es necesario que la anotación asociada implemente el método `setCoordinate:`

```
@property (nonatomic) MKAnnotationViewDragState dragState
```

- Contiene el estado del desplazamiento.
- Ver documentación para detalles.

. . .

MKAnnotationView

- Podemos crear nuestras propias views personalizadas de dos formas:
  - Modificando directamente el valor de las propiedades definidas en la propia clase `MKAnnotationView`
    - `image`, `leftCalloutAccessoryView`, ...
  - Creando una clase nueva derivada de `MKAnnotationView`.

Crear un objeto MKAnnotationView

- Los objetos `MKAnnotationView` se obtienen de forma similar a como se obtenían las `UITableViewCell` de las `UITableView`.
  - `MKMapView` usa un cola de reutilización para guardar los objetos que no se ven.
    - También se etiquetan con un identificador de reutilización.
  - Los objetos se sacan de está cola, y si la cola está vacía se crean objetos nuevos.
- Una vez que tenemos un objeto de tipo MKAnnotationView podemos
  - asignarle una imagen o pintar su contenido:
    `image, -drawRect:`
  - añadirle `UIControls` en sus accesorios:
    `leftCalloutAccessoryView`
    `rightCalloutAccessoryView`
  - y configurar otras propiedades:
    `centerOffset`, `calloutOffset`, `enabled`, `selected`, ...

```
- (MKAnnotationView *) mapView:(MKMapView *)mapView
            viewForAnnotation:(id <MKAnnotation>)annotation {

    static NSString* avid = @"DEST_AV_ID";

    if ([annotation isKindOfClass:[MKUserLocation class]]) {
        return nil;
    }

    MKPinAnnotationView * av = (MKPinAnnotationView *)
                    [mapView dequeueReusableAnnotationViewWithIdentifier:avid];
    if (av == nil) {
        av = [[MKPinAnnotationView alloc] initWithAnnotation:annotation
                                                  reuseIdentifier:avid];
        av.canShowCallout = YES;
        av.pinColor = MKPinAnnotationColorRed;

        UIButton * b = [UIButton buttonWithType:UIButtonTypeRoundedRect];
        b.bounds = CGRectMake(0,0,80,28);
        [b setTitle:@"delete" forState:UIControlStateNormal];
        av.leftCalloutAccessoryView = b;

        av.rightCalloutAccessoryView =
                    [UIButton buttonWithType:UIButtonTypeDetailDisclosure];
    }
    av.annotation = annotation;

    // Configurar aqui las cosas propias de esta anotacion.

    return av;
}
```

Callouts: Configuración Perezosa

- Consejo: Configurar de forma perezosa los callouts.
  - El callout de una anotación sólo se muestra cuando se selecciona su annotationView.
  - Y entonces se llama al método mapView:didSelectAnnotationView: del delegado
  - Este método nos permite configurar de forma perezosa el callout:
    - Configuramos el callout sólo cuando va a mostrarse el callout en pantalla.
      - Por ejemplo: En este momento descargaremos las imágenes a mostrar en las accesory views. Las imágenes de los callout que no se muestren no se descargarán.

Overlays

Overlays

- Son anotaciones que además de definir una coordenada, también definen un área del mapa.
- Permiten representar datos usando un layer por encima del mapa.
  - Ejemplos: una ruta entre dos puntos, los límites de un campo de golf, ...
- Se manejan usando unos mecanismos similares a los usados con la anotaciones.

MKOverlay

- El protocolo `MKOverlay` incluye `MKAnnotation` y además define:
  - Rectángulo que contiene el overlay.

```
@property (readonly) MKMapRect boundingMapRect;
```

  - Para saber cuando debe pintarse el overlay:

```
-(BOOL)intersectsMapRect:(MKMapRect)mapRect;
```

    - Es opcional.
    - Si se omite se usa `boundingMapRect`.

Añadir a un mapa

- Los overlays se añaden o quitan de un mapa usando:
  - `- (void)addOverlay:(id<MKOverlay>)overlay`
  - `- (void)addOverlays:(NSArray *)overlays`
  - `- (void)removeOverlay:(id<MKOverlay>)overlay`
  - `- (void)removeOverlays:(NSArray *)overlays`

MKOverlayView

- Los overlays se muestran en el mapa usando un objeto `MKOverlayView`.
- El delegado del mapa proporciona estos objetos mediante el método:

```
-(MKOverlayView*)mapView:(MKMapView*)mapView
          viewForOverlay:(id<MKOverlay>)overlay
```

- Los objetos `MKOverlayView` se crean usando:

```
-(MKOverlayView*)mapView:(MKMapView*)mapView
          viewForOverlay:(id<MKOverlay>)overlay
```

  - No modificar el `frame` de esta view, es decir, su posición o tamaño. El mapa lo gestiona cuando necesita repintar.

- Para pintar el contenido del overlay hay que crearse una subclase de `MKOverlayView` y sobreescribir el método:

```
-(void)drawMapRect:(MKMapRect)mapRect
         zoomScale:(MKZoomScale)zoomScale
         inContext:(CGContextRef)context
```

  - Los overlays pueden dibujarse desde varios threads. Este método debe ser **thread-safe**.
  - El área a dibujar se especifica usando las coordenadas del mapa. Usar estas coordenadas para calcular que hay que dibujar, y luego convertirlas en coordenadas de la view.
  - Ya nos pasan el contexto gráfico a usar para dibujar el contenido de la view.

- Métodos de `MKOverlayView` para hacer conversiones entre coordenadas del mapa y puntos de la view

```
-(MKMapPoint)mapPointForPoint:(CGPoint)point;
-(MKMapRect)mapRectForRect:(CGRect)rect;
-(CGPoint)pointForMapPoint:(MKMapPoint)mapPoint;
-(CGRect)rectForMapRect:(MKMapRect)mapRect;
```

MKReverseGeocoder

MKReverseGeocoder

- Deprecado en iOS 5.
  - Usar: `CLGeocoder`.
- Dada una localización, obtener su dirección postal.
- Crear objeto con:

```
-(id) initWithCoordinate:
                (CLLocationCoordinate2D)coord
```

- Arrancar y parar la geolocalización:

`-start` y `-cancel`

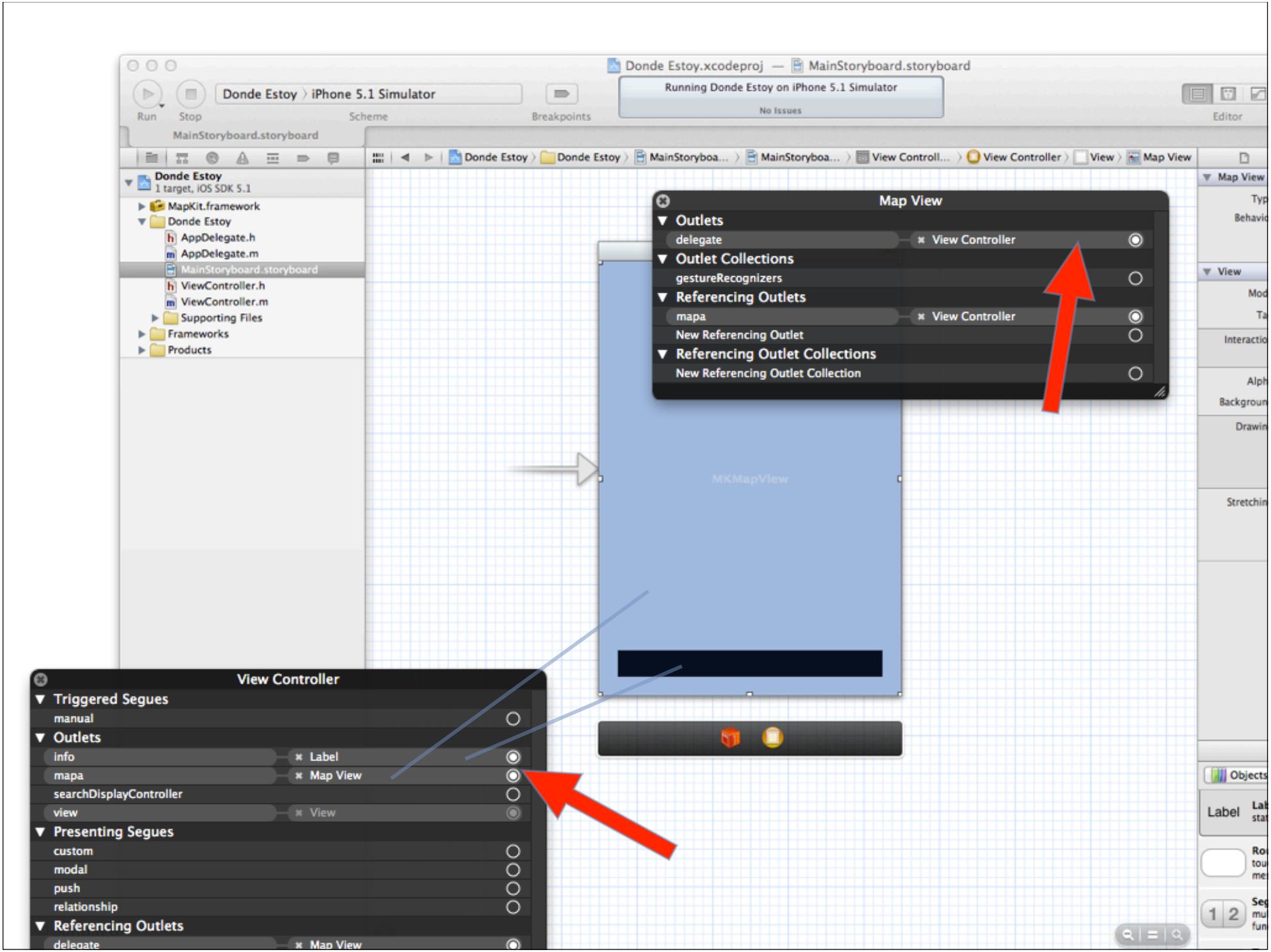
- Al terminar se avisa a su delegado:

```
–reverseGeocoder:didFindPlacemark:
–reverseGeocoder:didFailWithError:
```

Demo - ¿Donde Estoy?

Demo - ¿Donde Estoy?

- Crear una app basada en un ViewController.
  - Añadir el framework MapKit.
- Editar el Storyboard para añadir al ViewController:
  - Un objeto MKMapView para mostrar el mapa.
    - Asignar el ViewController como delegado del mapa.
      - El ViewController debe ser conforme a MKMapViewDelegate.
  - Un objeto UILabel para mostrar la longitud y la latitud
  - Crear outlets al mapa y la etiqueta.


Donde Estoy.xcodeproj — MainStoryboard.storyboard
Donde Estoy 〉iPhone 5.1 Simulator
Running Donde Estoy on iPhone 5.1 Simulator
No Issues
Run
Stop
Scheme
Breakpoints
Editor
MainStoryboard.storyboard
Donde Estoy
1 target, iOS SDK 5.1
MapKit.framework
Donde Estoy
AppDelegate.h
AppDelegate.m
MainStoryboard.storyboard
ViewController.h
ViewController.m
Supporting Files
Frameworks
Products
Map View
Outlets
delegate
View Controller
Outlet Collections
gestureRecognizers
Referencing Outlets
mapa
View Controller
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
MKMapView
View Controller
Triggered Segues
manual
Outlets
info
Label
mapa
Map View
searchDisplayController
view
View
Presenting Segues
custom
modal
push
relationship
Referencing Outlets
delegate
Map View
Objects
Label

- Mostrar Teleco en modo híbrido (viewDidLoad):

```
MKCoordinateRegion reg;
reg.center.latitude = 40.452580;
reg.center.longitude = -3.726372;
reg.span.latitudeDelta = 0.003;
reg.span.longitudeDelta = 0.003;

reg = [self.mapa regionThatFits:reg];
self.mapa.region = reg;

self.mapa.mapType = MKMapTypeHybrid;
```

- Para refrescar el outlet info con las coordenadas del sitio visualizado en el mapa, en ViewController.h, que es el delegado del mapa, añado:

```
- (void) mapView:(MKMapView *)mapView
        regionDidChangeAnimated:(BOOL)animated {

    self.info.text = [NSString stringWithFormat:@"%f %f",
                      mapView.centerCoordinate.latitude,
                      mapView.centerCoordinate.longitude];
}
```

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

@interface ViewController () <MKMapViewDelegate>
@property (weak, nonatomic) IBOutlet MKMapView *mapa;
@property (weak, nonatomic) IBOutlet UILabel *info;
@end

@implementation ViewController
@synthesize mapa = _mapa;
@synthesize info = _info;

- (void)viewDidLoad
{
    [super viewDidLoad];

    MKCoordinateRegion reg; // = { {40.452580, -3.726372} , {0.003, 0.003} };

   reg.center.latitude = 40.452580;
   reg.center.longitude = -3.726372;
   reg.span.latitudeDelta = 0.003;
   reg.span.longitudeDelta = 0.003;

   reg = [self.mapa regionThatFits:reg];

   self.mapa.region = reg;
   self.mapa.mapType = MKMapTypeHybrid;
}


- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated
{
    self.info.text = [NSString stringWithFormat:@"%f %f",
                      mapView.centerCoordinate.latitude,
                      mapView.centerCoordinate.longitude];
}

...
@end
```

Uso privado del protocolo y outlets.

Demo - Crear un Ruta

- Queremos volver a Teleco desde nuestra posición actual.
- Hacer Long Press para indicar puntos intermedios por donde pasar.
- Podemos borrar los puntos intermedios.
- Nos muestra los kilómetros a recorrer.