



POLITÉCNICA

ETSIT
UPM

dit
UPM

Desarrollo de Apps para iOS Core Location

IWEB,LSWC 2013-2014

Santiago Pavón

ver: 2012.09.05 p1

Framework Core Location

- Manejar posición, movimiento, altitud, dirección ...
- Usa GPS, Wifi, antenas telefónicas.
 - Elevado consumo de batería
 - A mayor precisión, mayor gasto.
 - Consumo GPS > Wifi > telefonía
- Seguridad:
 - Se pide autorización al usuario.
 - Existe una lista negra de aplicaciones.
- Incluir el framework **CoreLocation** al proyecto.
 - y añadir

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

CLLocation

- Los objetos de la clase CLLocation contienen los datos de localización que ha generado un objeto de la clase CLLocationManager.

- Propiedades:

```
CLLocationCoordinate2D coordinate;  
CLLocationDistance altitude;  
CLLocationAccuracy horizontalAccuracy;  
CLLocationAccuracy verticalAccuracy;  
CLLocationDirection course;  
CLLocationSpeed speed;  
NSDate * timestamp;
```

- Métodos:

```
-(CLLocationDistance) getDistanceFrom:(const CLLocation*) location;
```

. . .

CLLocationHeading

- Los objetos de la clase CLLocationHeading contienen los datos de orientación que ha generado un objeto de la clase CLLocationManager.
- Propiedades:

```
CLLocationDirection    magneticHeading;  
CLLocationDirection    trueHeading;  
CLLocationDirection    headingAccuracy;  
CLLocationHeadingComponentValue x;  
CLLocationHeadingComponentValue y;  
CLLocationHeadingComponentValue z;  
NSDate *               timestamp;
```

- Métodos:

```
-(NSString*) description;
```

CLLocationManager

- Los objetos de esta clase se configuran para establecer cuando queremos que nos informen de la localización y orientación actual del terminal.
 - Informan de forma asíncrona de los cambios de localización y orientación
 - posición, altitud, orientación, precisión, fecha, ...
 - Informan usando su objeto delegado.
 - Un objeto de la clase **CLLocationManagerDelegate**.
- Hay que configurar cuando queremos que nos informen:
 - precisión deseada.
 - Recordad: a mayor precisión, mayor gasto de batería.
 - filtro de distancia.
 - no informa de movimientos inferiores a esa distancia.
 - cambios de región.
 - cambios de orientación.
 - etc.
- También se puede preguntar a estos objetos en cualquier momento cual es la localización u orientación actual.

Servicios de Localización Disponibles

- La clase `CLLocationManager` dispone de métodos que:
 - indican cuales son los servicios de localización que soporta el hardware del terminal.
 - + `significantLocationChangeMonitoringAvailable`
 - + `headingAvailable`
 - + `regionMonitoringAvailable`
 - indican que servicios están habilitados
 - + `locationServicesEnabled`
 - + `regionMonitoringEnabled`
 - indican si se tiene autorización del usuario para usar estos servicios
 - + `authorizationStatus`

Tipos de Monitorización

- CLLocationManager puede monitorizar e informar de diferentes tipos de cambios:
 - Cambios en la localización.
 - basándose en medidas continuas de la posición actual y usando la precisión y filtro de distancia configurado.
 - Cambios significativos de la localización.
 - Bajo consumo de energía.
 - Poca precisión.
 - Se informa aunque la aplicación no se esté ejecutando, o esté en background.
 - Cambios de región.
 - Se informa al cambiar de región geográfica.
 - Se pueden configurar las regiones y sus fronteras.
 - Se informa aunque la aplicación no se esté ejecutando, o esté en background
 - Cambios de orientación.
 - Se realizan medidas continuas de la orientación actual y se informa según el filtro de orientación configurado.

Propiedades y Métodos

- Cambios de la localización:

- (void) **startUpdatingLocation**

- (void) **stopUpdatingLocation**

- CLLocationAccuracy **desiredAccuracy**

- CLLocationDistance **distanceFilter**

- Cambios de la orientación:

- (void) **startUpdatingHeading**

- (void) **stopUpdatingHeading**

- (void) **dismissHeadingCalibrationDisplay**

- CLLocationDegrees **headingFilter**

- CLDeviceOrientation **headingOrientation**

Propiedades y Métodos

- Cambios significativos de la localización:
 - (void) **startMonitoringSignificantLocationChanges**
 - (void) **stopMonitoringSignificantLocationChanges**

- Cambios de región:
 - (void) **startMonitoringForRegion:**
 - (void) **startMonitoringForRegion:desiredAccuracy:**
 - (void) **stopMonitoringForRegion:**

NSSet * monitoredRegions

CLLocation Distance maximumRegionMonitoringDistance

Propiedades y Métodos

- Obtener última localización y orientación:

```
CLLocation * location  
CLHeading * heading
```

- Describir al usuario el motivo de querer usar el servicio de localización:

```
NSString * purpose
```

- Acceso al delegado

```
id<CLLocationManagerDelegate> delegate
```

CLLocationManagerDelegate

- (void) **locationManager:** (CLLocationManager*)manager
 didUpdateToLocation: (CLLocation*)newLocation
 fromLocation: (CLLocation*)oldLocation;
- (void) **locationManager:** (CLLocationManager*)manager
 didUpdateHeading: (CLHeading*)newHeading;
- (BOOL) **locationManagerShouldDisplayHeadingCalibration:**
 (CLLocationManager *)manager;

CLLocationManagerDelegate

- (void) **locationManager:** (CLLocationManager*)manager
 didFailWithError: (NSError*)error;

- Consultar el código de error de NSError para identificar el tipo de fallo ocurrido:

kCLErrorLocationUnknown

- No se ha podido obtener la localización.
- Se sigue intentando.

kCLErrorDenied

- El usuario no permite que la aplicación acceda a Core Location.

kCLErrorHeadingFailure

- Demasiadas interferencias magnéticas.
- Se sigue intentando.

CLLocationManagerDelegate

- (void) **locationManager:** (CLLocationManager*) manager
 didEnterRegion: (CLRegion*) region
- (void) **locationManager:** (CLLocationManager*) manager
 didExitRegion: (CLRegion*) region
- (void) **locationManager:** (CLLocationManager*) manager
monitoringDidFailForRegion: (CLRegion*) region
 withError: (NSError*) error
- (void) **locationManager:** (CLLocationManager*) manager
 didStartMonitoringForRegion: (CLRegion*) region

CLLocationManagerDelegate

```
- (void) locationManager:(CLLocationManager *)manager  
  didChangeAuthorizationStatus:(CAuthorizationStatus)status
```

Demo: ¿Donde Estoy?

- Propiedades:

```
CLLocationManager * locMgr;  
IBOutlet UILabel * latitudLabel;
```

- En viewDidLoad

```
self.locMgr = [[CLLocationManager alloc] init];  
self.locMgr.delegate = self;
```

- En viewWillAppear

```
[self.locMgr startUpdatingLocation];
```

- Delegado > locationManager:didUpdateToLocation:fromLocation:

```
CLLocationDegrees latitud = newLocation.coordinate.latitude;  
self.latitudLabel.text = [NSString stringWithFormat:@"%f°", latitud];
```

- Ver detalladamente el código del ejemplo.

CLGeocoder

- Forward geocoding:
 - Dada una dirección, obtener las coordenadas.
- Reverse geocoding:
 - Dadas unas coordenadas, obtener una dirección.
- Demo: Consultar el ejemplo "Ruta a Teleco".
 - Al mostrar el callout de las anotaciones del mapa (sitios por donde pasa la ruta) se usa reverse geocoding para obtener su dirección.
 - Es una evaluación perezosa: sólo realizamos esta operación la primera vez que se muestra el callout. Si no se muestra no se hace nada.

