

Ya empieza:

iBeacons

micro locations



iBeacons

micro locations





Ezequiel Aceto

- CTO de Everypost
- Desarrollador mobile desde...bueno... Palm OS
- Twitter: @eaceto



- iBeacon
- Bluetooth
- Integración en iOS
- Background



- Indoor LBS
- Location aware
- NFC
- Bluetooth LE



micro locations

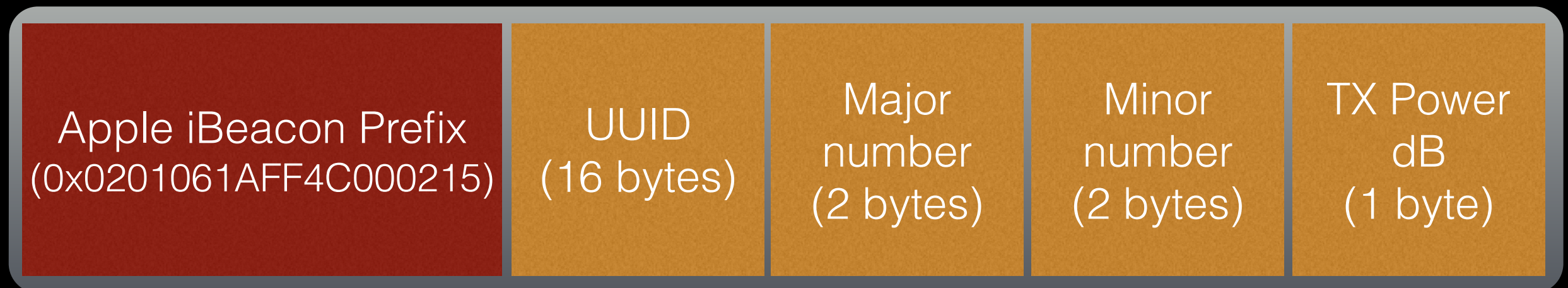


- Es la tecnología en la que se basa iBeacon
- Muy bajo consumo
- Scanning Mode, Master / Slave
- Advertising Mode



- Cabecera: 0xAA (siempre)
- Dirección de acceso al canal RF: 0x8E89BED6
- PDU: Entre 2 y 39 bytes
- CRC: 3 bytes sobre PDU





- iPhone 4S o superior
- iPad 3 o superior
- Cualquier iPad mini
- iPod Touch 5th gen
- Cualquier Mac con OSX 10.9 y Bluetooth 4.0
- Android 4.3+ con BT LE



¿Cómo usamos iBeacons en iOS 7?





CoreLocation



CoreBluetooth



```
@interface CLBeacon : NSObject
```

```
//Proximity identifier associated with the beacon.  
@property (readonly, nonatomic) NSUUID *proximityUUID;  
//Most significant value associated with the beacon.  
@property (readonly, nonatomic) NSNumber *major;  
//Least significant value associated with the beacon.  
@property (readonly, nonatomic) NSNumber *minor;  
//Proximity of the beacon from the device.  
@property (readonly, nonatomic) CLProximity proximity;  
//Received signal strength in decibels of the specified beacon.  
@property (readonly, nonatomic) NSInteger rssi;
```

```
@interface CLBeaconRegion : CLRegion
```

```
// Proximity identifier associated with the region.  
@property (readonly, nonatomic) NSUUID *proximityUUID;  
//Most significant value associated with the region. If a major value wasn't  
specified, this will be nil.  
@property (readonly, nonatomic) NSNumber *major;  
//Least significant value associated with the region. If a minor value wasn't  
specified, this will be nil.  
@property (readonly, nonatomic) NSNumber *minor;
```



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

```
-(void)startLookingForBeacons {
```

```
    NSUUID *myBeaconUUID = [[NSUUID alloc]
                             initWithUUIDString:@"E3EAC100-1108-1984-
AB1A-255255255E1A"];
```

```
    CLBeaconRegion *region = [[CLBeaconRegion alloc]
                              initWithProximityUUID:myBeaconUUID
                              identifier:@"A Beacon Region"];
```

```
    [self.locationManager startRangingBeaconsInRegion:region];
```

```
}
```




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

```
-(void)startLookingForBeacons {
```

```
    NSUUID *myBeaconUUID = [[NSUUID alloc]
                             initWithUUIDString:@"E3EAC100-1108-1984-
AB1A-255255255E1A"];
```

```
    CLBeaconRegion *regionA = [[CLBeaconRegion alloc]
                                initWithProximityUUID:myBeaconUUID
                                major:[NSNumber numberWithInt:127]
                                identifier:@"First Beacon Region"];
```

```
    CLBeaconRegion *regionB = [[CLBeaconRegion alloc]
                                initWithProximityUUID:myBeaconUUID
                                major:[NSNumber numberWithInt:127]
                                minor:[NSNumber numberWithInt:1]
                                identifier:@"Second Beacon Region"];
```

```
[self.locationManager startRangingBeaconsInRegion:regionA];
[self.locationManager startRangingBeaconsInRegion:regionB];
```

```
}
```



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

```
-(void) locationManager:(CLLocationManager *)manager  
    didRangeBeacons:(NSArray *)beacons  
    inRegion:(CLBeaconRegion *)region {
```

```
    CLBeacon *beacon = [beacons firstObject];
```

```
    NSLog(@"proximityUUID: %@", beacon.proximityUUID.UUIDString);
```

```
    NSLog(@"Major: %@", beacon.major);
```

```
    NSLog(@"Minor: %@", beacon.minor);
```

```
    NSLog(@"RSSI: %i dB", beacon.rssi);
```

```
    if (beacon.proximity == CLProximityUnknown) {
```

```
        NSLog(@"Unknown Proximity");
```

```
    } else if (beacon.proximity == CLProximityImmediate) {
```

```
        NSLog(@"Immediate");
```

```
    } else if (beacon.proximity == CLProximityNear) {
```

```
        NSLog(@"Near");
```

```
    } else if (beacon.proximity == CLProximityFar) {
```

```
        NSLog(@"Far");
```

```
    }
```

```
}
```



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

```
-(void)locationManager:(CLLocationManager *)manager  
    didEnterRegion:(CLRegion *)region {
```

```
    CLBeaconRegion *ibr = [[CLBeaconRegion alloc]  
                           initWithProximityUUID:myBeaconUUID  
                           identifier:@"My iBeacon"];
```

```
    [self.locationManager startRangingBeaconsInRegion:ibr];
```

```
}
```

```
-(void)locationManager:(CLLocationManager *)manager  
    didExitRegion:(CLRegion *)region {
```

```
    CLBeaconRegion *ibr = [[CLBeaconRegion alloc]  
                           initWithProximityUUID:myBeaconUUID  
                           identifier:@"My iBeacon"];
```

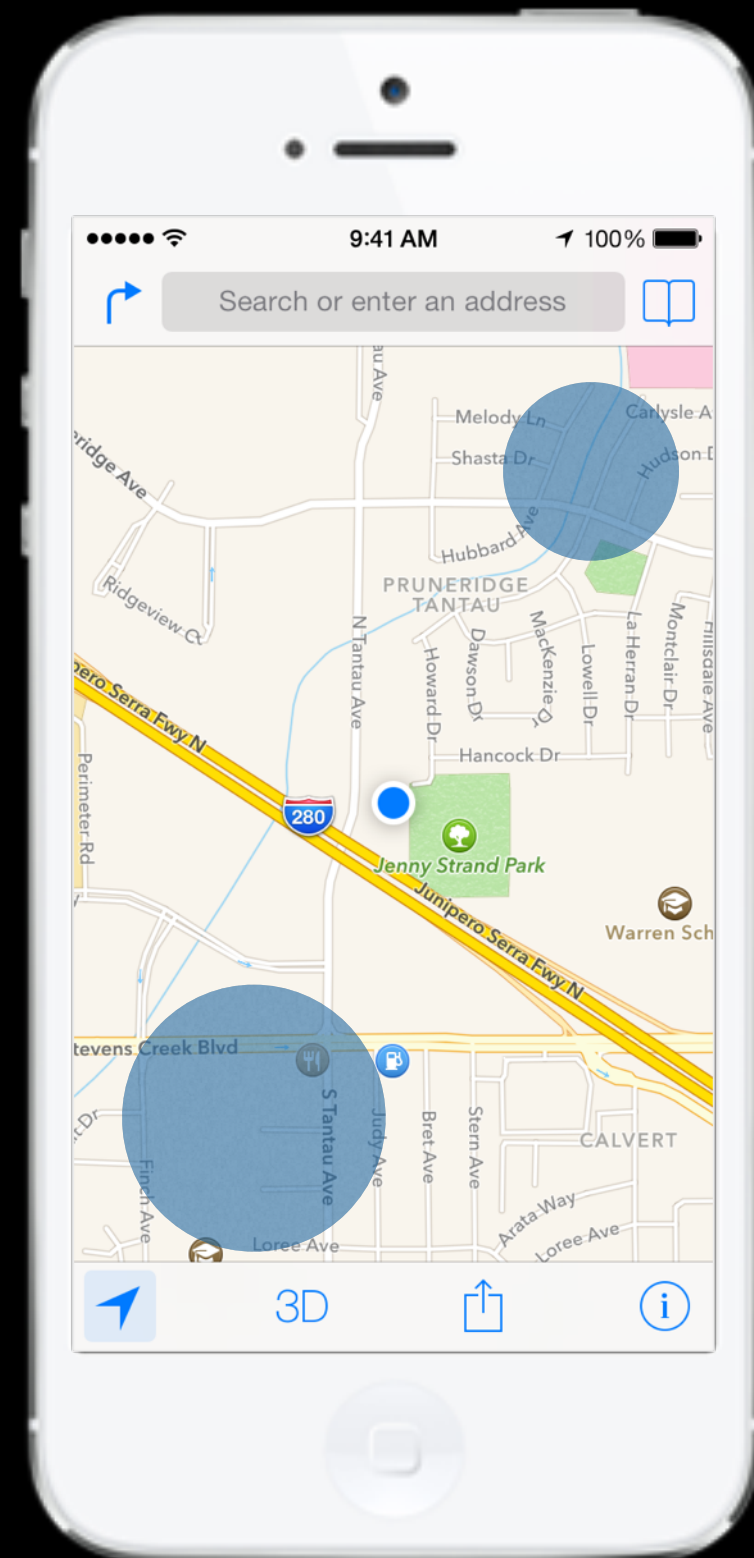
```
    [self.locationManager stopRangingBeaconsInRegion:ibr];
```



- Si monitoreamos regiones, el evento se produce al cruzar la región.
 - Entonces usamos `requestStateForRegion:`
- Las regiones son un recurso compartido
- Solo podemos registrar 20 regiones por App
- Agrupar varios iBeacons en una región



- Utilizar CoreLocation
- Definir regiones de “GPS”





- Maximizar velocidad de respuesta

```
[locationManager startMonitoringForRegion:beaconRegion];  
[locationManager startRangingBeaconsInRegion:beaconRegion];
```
- ¿Y cuando el usuario (re)inicia su dispositivo?

```
beaconRegion.notifyEntryStateOnDisplay = YES;
```
- No olvidar Background Location = YES



[github.com/eaceto/
NSCConfAR-iBeacon](https://github.com/eaceto/NSCConfAR-iBeacon)



¿Preguntas?



¡Gracias!



iBeacons



ezequielaceto.com.ar

@eaceto