

Maps

06.04.2020 - Swift 5



Introdução

- O que preciso para criar um mapa?
- O que podemos fazer com ele?
- Devemos solicitar permissão?
- Qual a implementação correta?



MapKit é um framework usado para mostrar um mapa em uma view dentro da aplicação. Pode ser usado para mostrar pontos de interesse, determinar informações em lugares e coordenadas.

MapKit

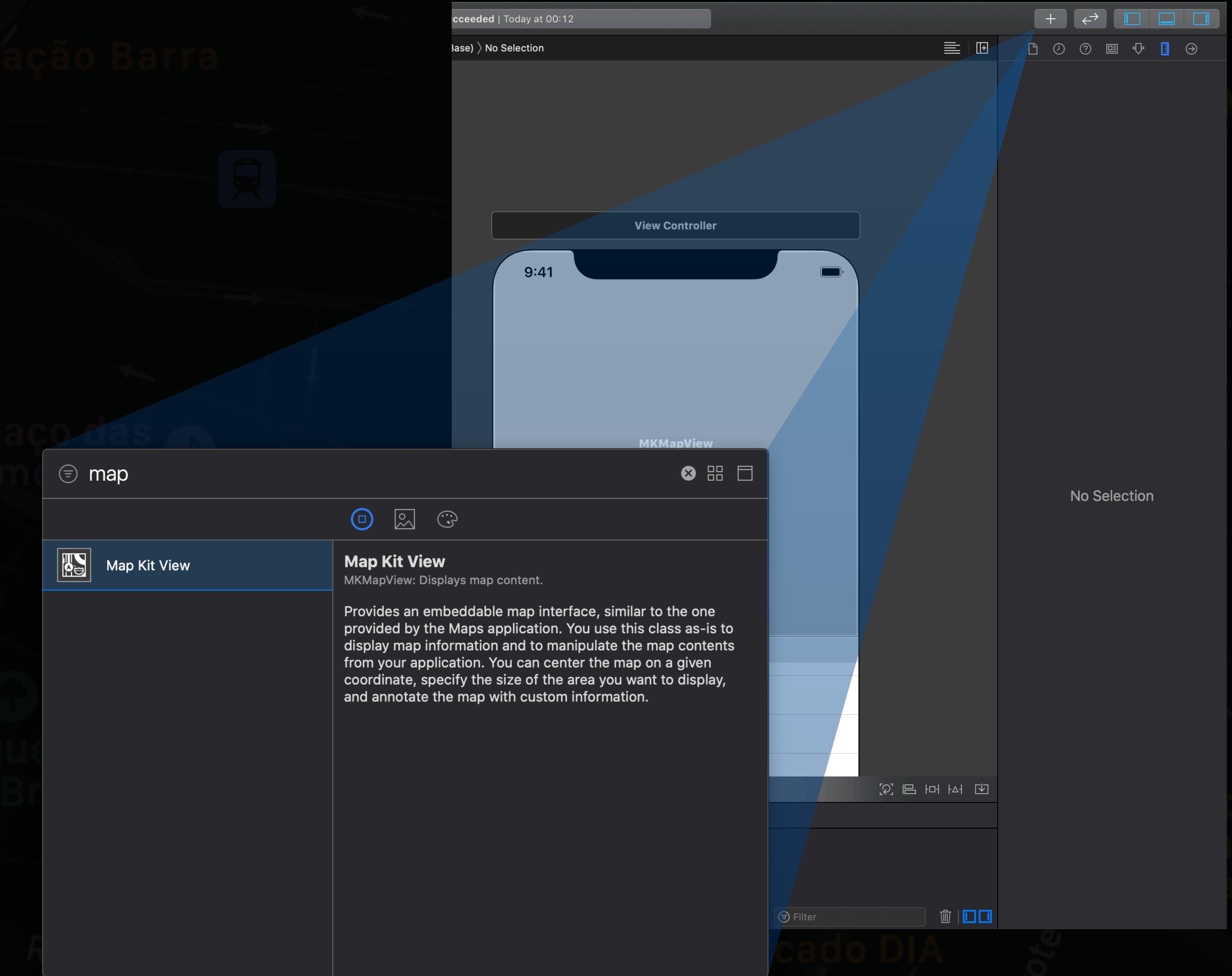
Framework

- Map View
- User Location
- Annotation
- Overlay
- Directions and Route
- Local Search
- Map Camera

MapKit

MKMapView

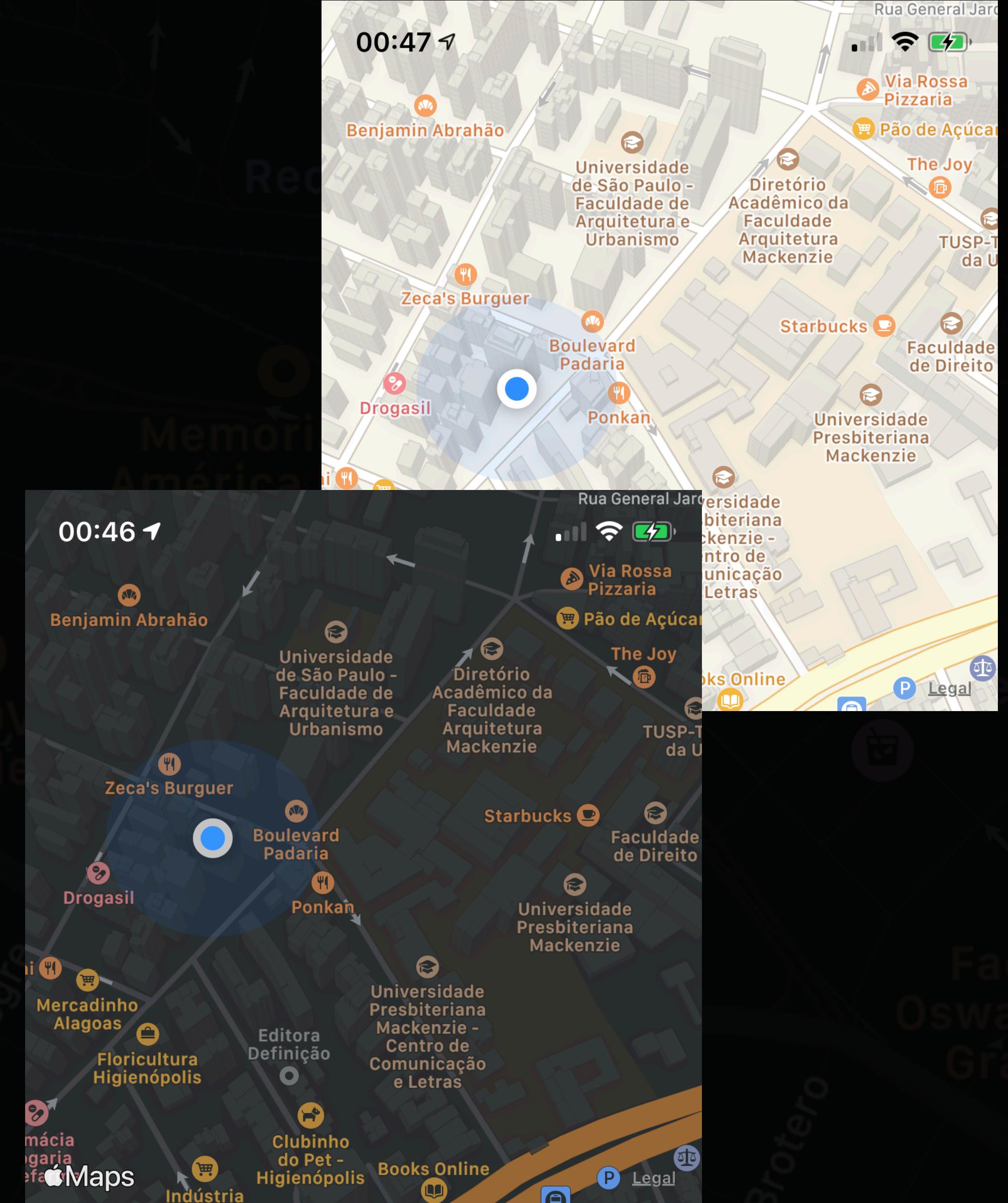
O Map Kit View é um objeto que prove a interface para incorporar o mapa na aplicação



MapKit

User Location

Essa propriedade permite exibir o local onde o usuário está no mapa. Mas para exibir a localização é necessário usar o Core Location Framework e autorizar a coleta da geolocalização do usuário.



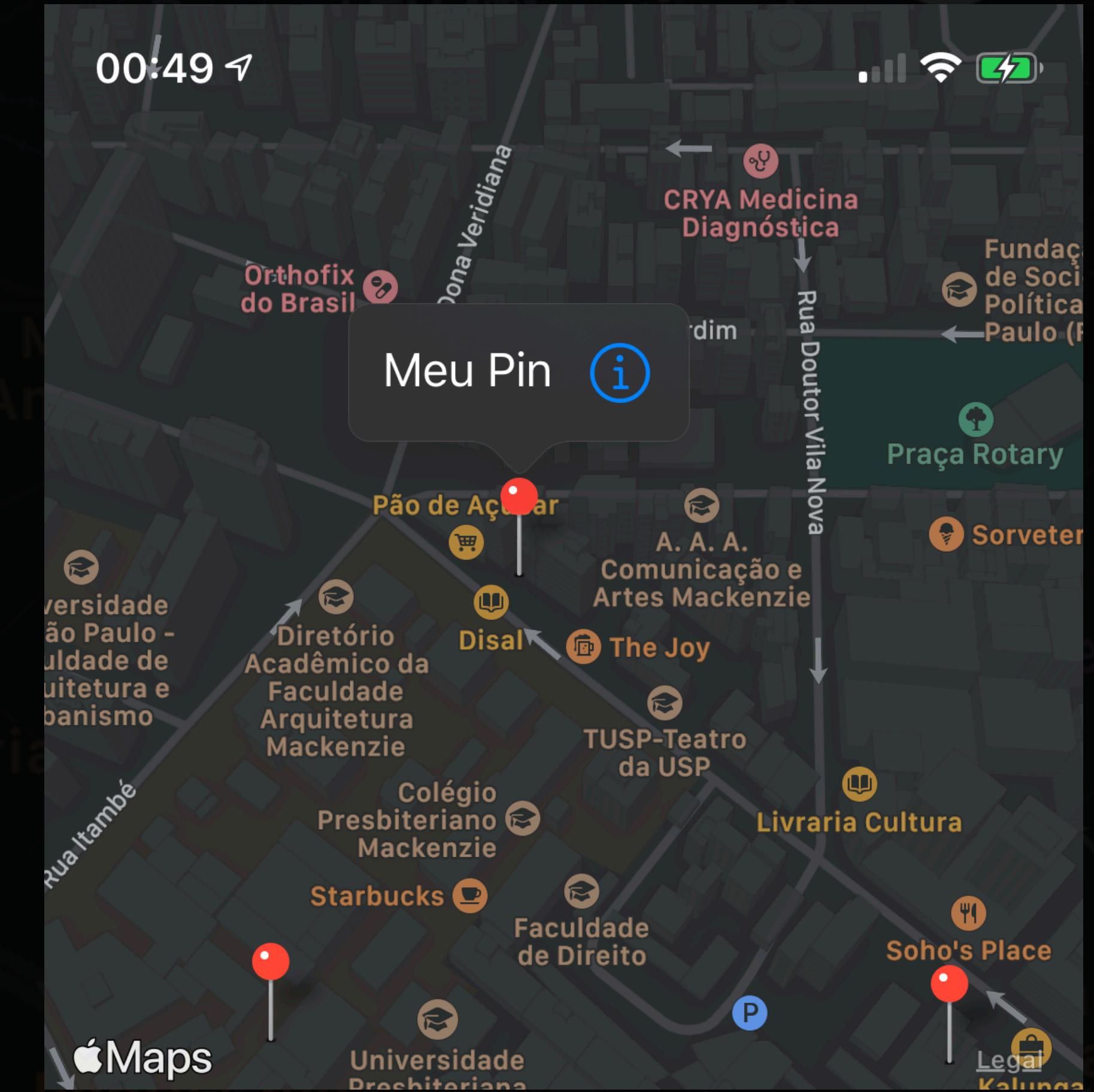
User Location

Exemplo

```
@IBOutlet var map: MKMapView!  
  
var locationManager = CLLocationManager()  
  
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    locationManager.delegate = self  
}  
  
@IBAction func showUserLocation(_ sender: AnyObject) {  
    map.showsUserLocation = !map.isUserLocationVisible  
}  
  
@IBAction func requestUserAuthorization(_ sender: AnyObject) {  
    locationManager.requestWhenInUseAuthorization()  
}
```

Anotações MKAnnotationView

A classe MKAnnotationView é responsável por mostrar anotações visuais no mapa. Elas correspondem a uma coordenada em um ponto na região demarcada.



MKAnnotationView

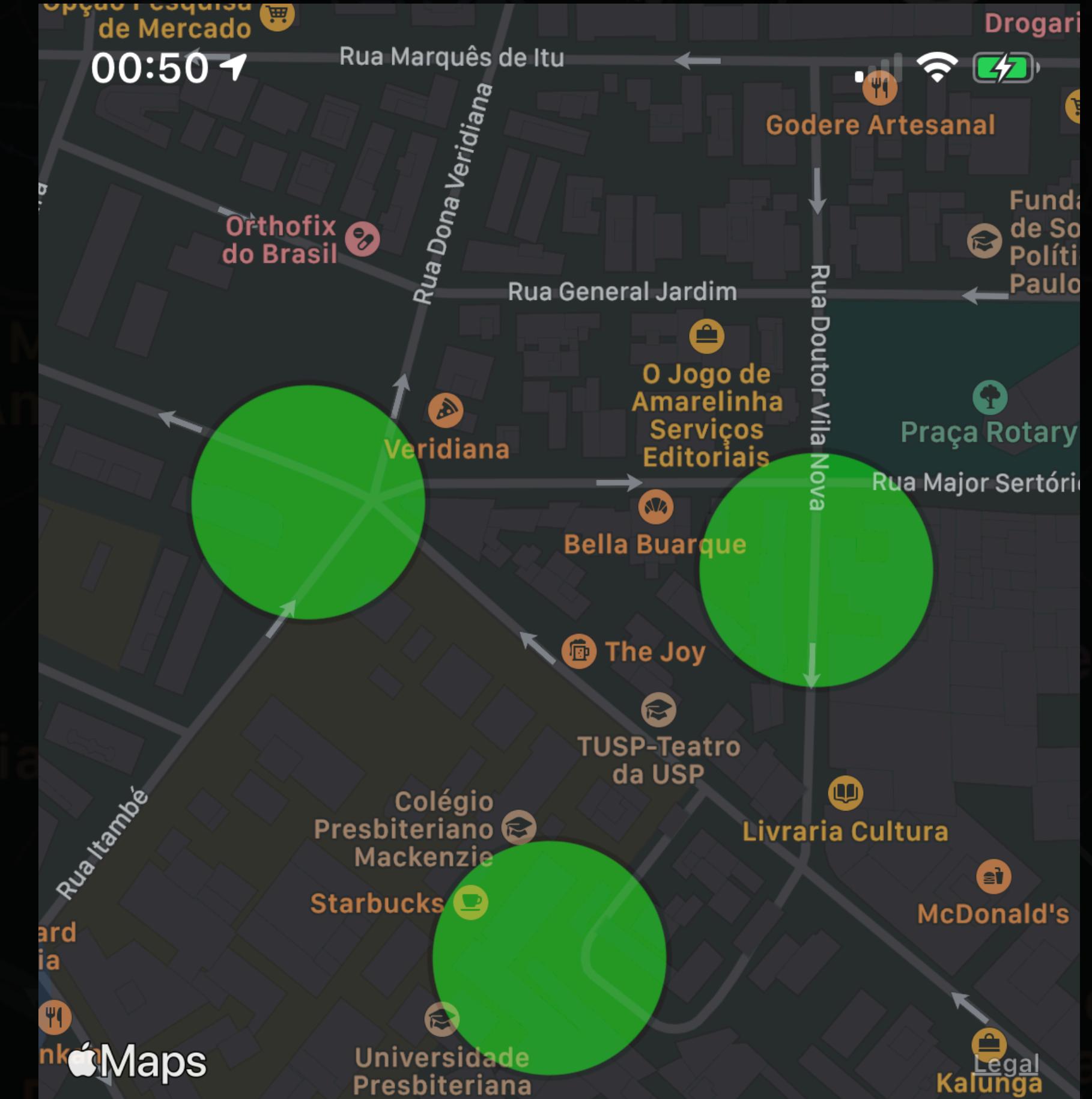
Exemplo

```
override func touchesEnded(_ touches: Set<UITouch>, with event: UIEvent?) {  
    if let touch = touches.first {  
        let position :CGPoint = touch.location(in: view)  
  
        let ann = MKPointAnnotation()  
        ann.coordinate = map.convert(position, toCoordinateFrom: self.view)  
        ann.title = "Meu Pin"  
  
        map.addAnnotation(ann)  
    }  
}
```

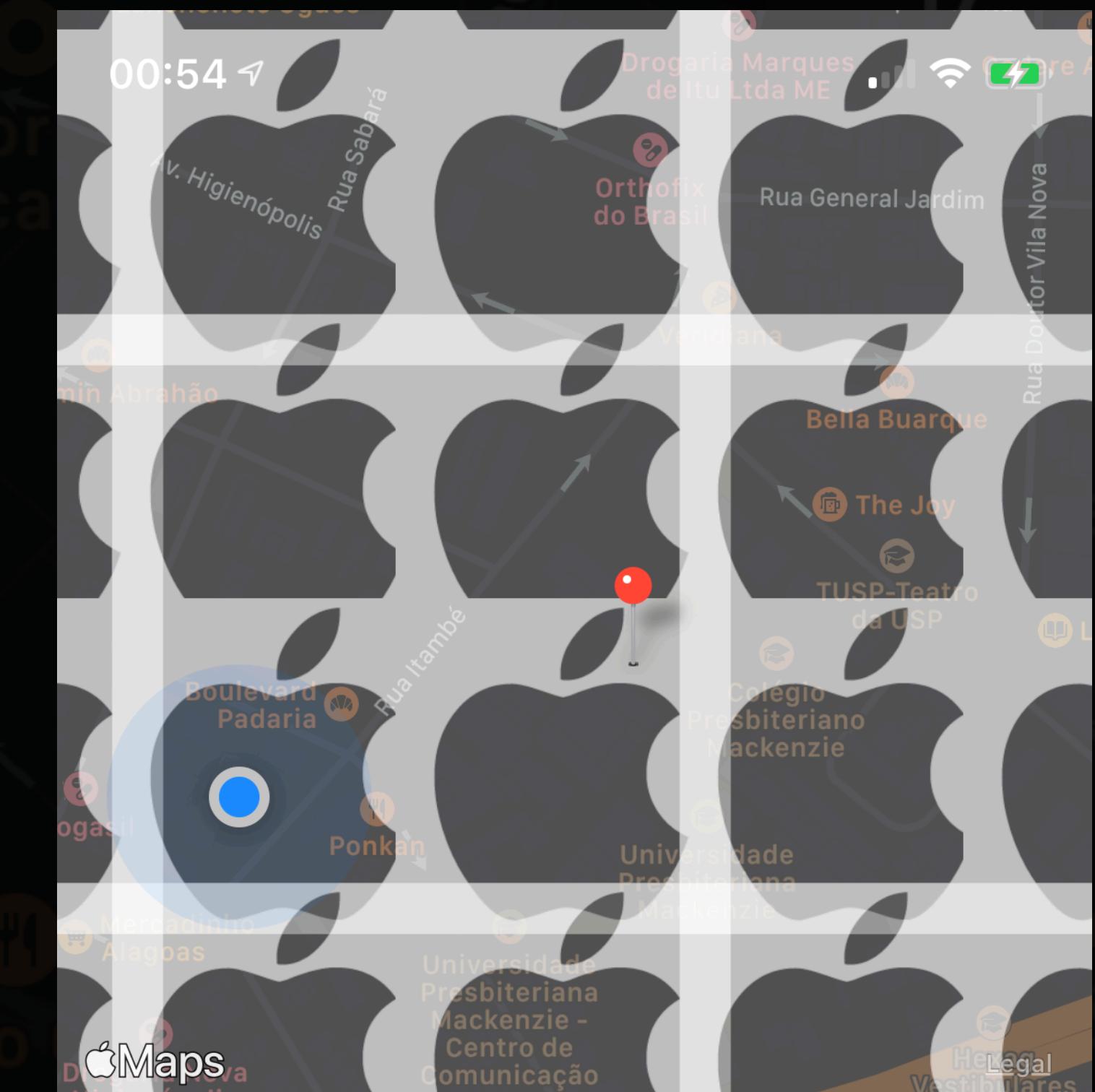
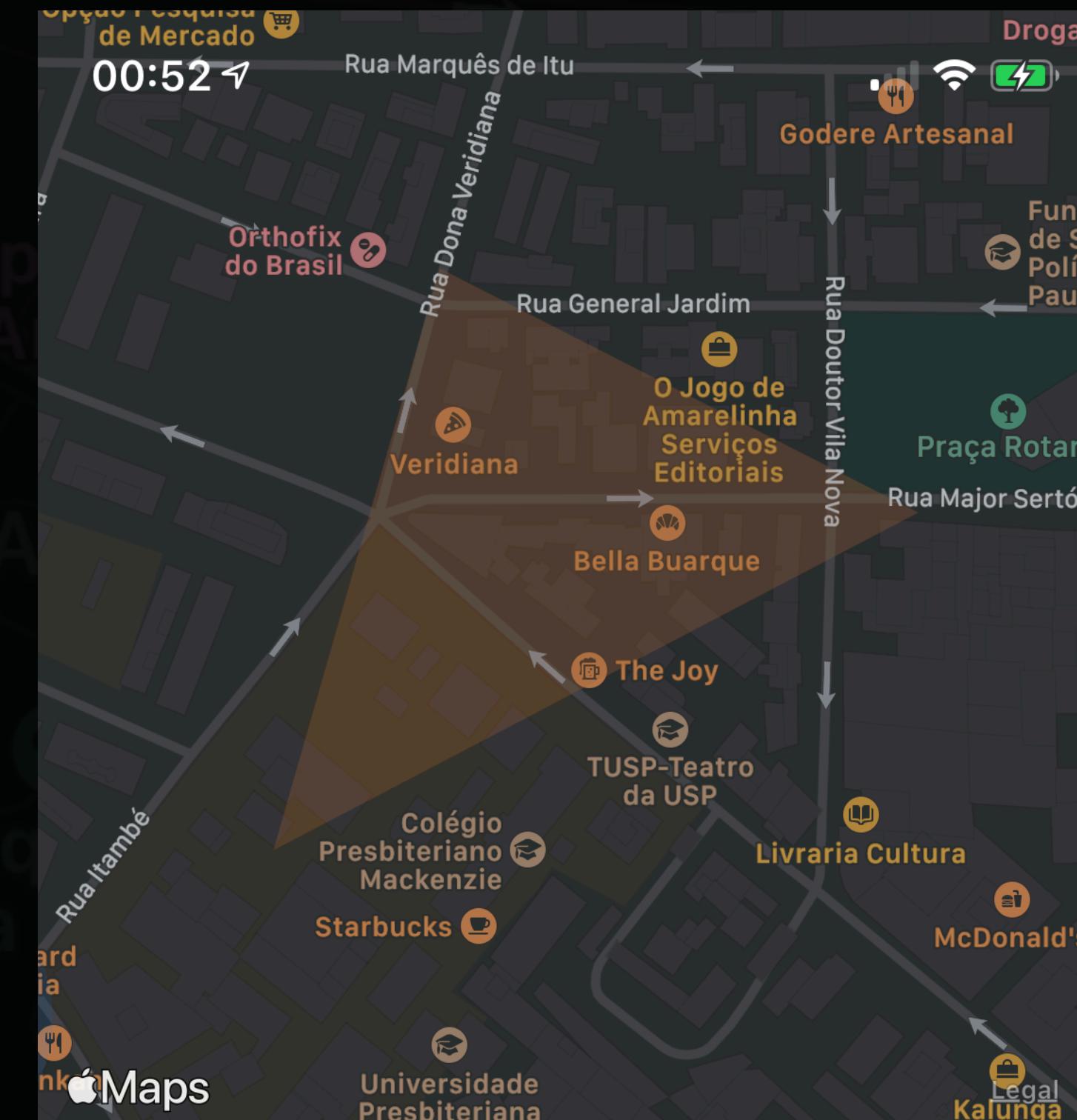
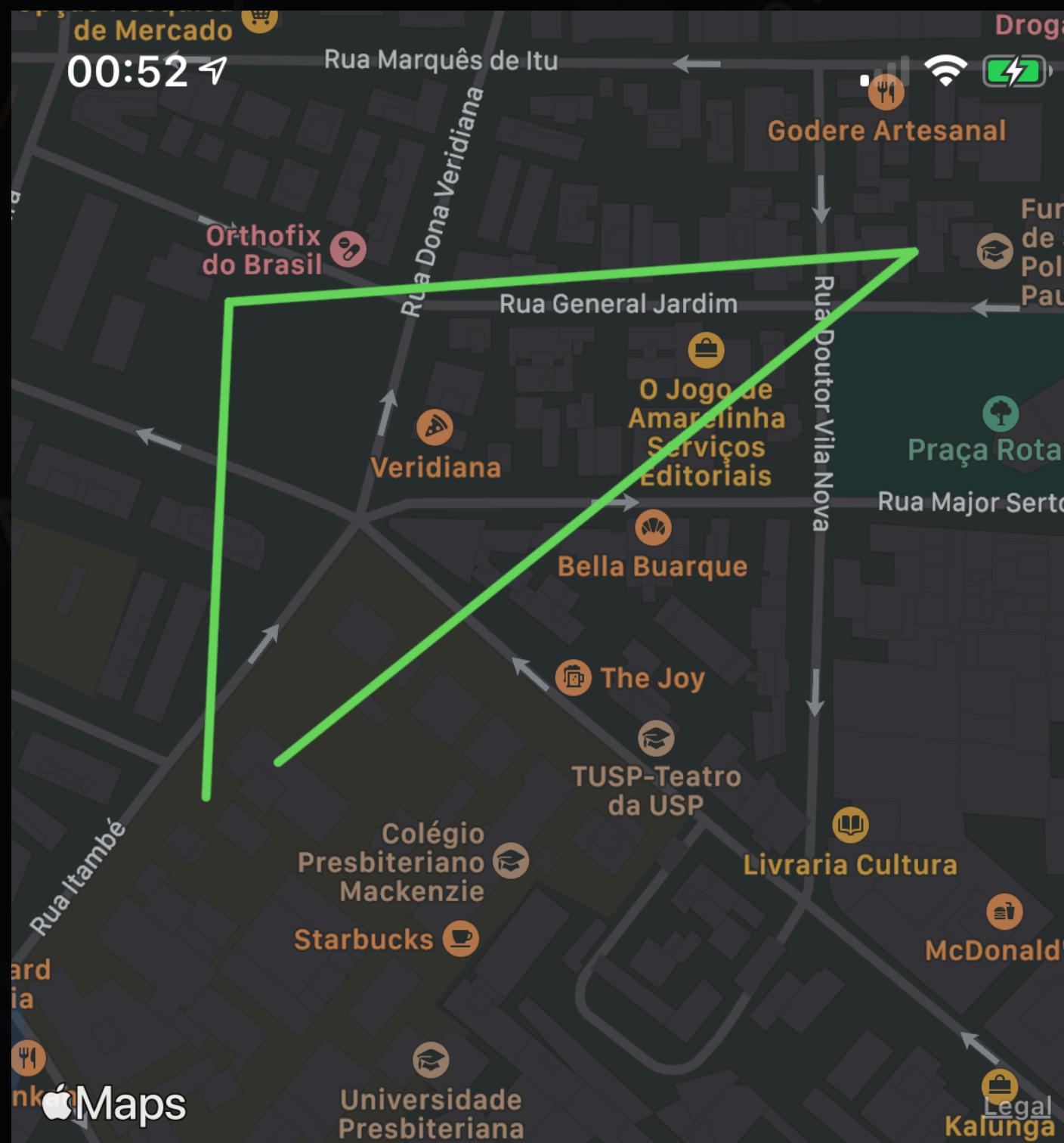
Sobreposição MKOverlay

A classe MKOverlayView prove a possibilidade de demarcar no mapa uma região de forma customizada. Para criar essa sobreposição no mapa, usa-se os seguintes objetos:

- MKCircle
- MKPolygon
- MKPolyline
- MKTileOverlay



Sobreposição MKOverlay



MKOverlay

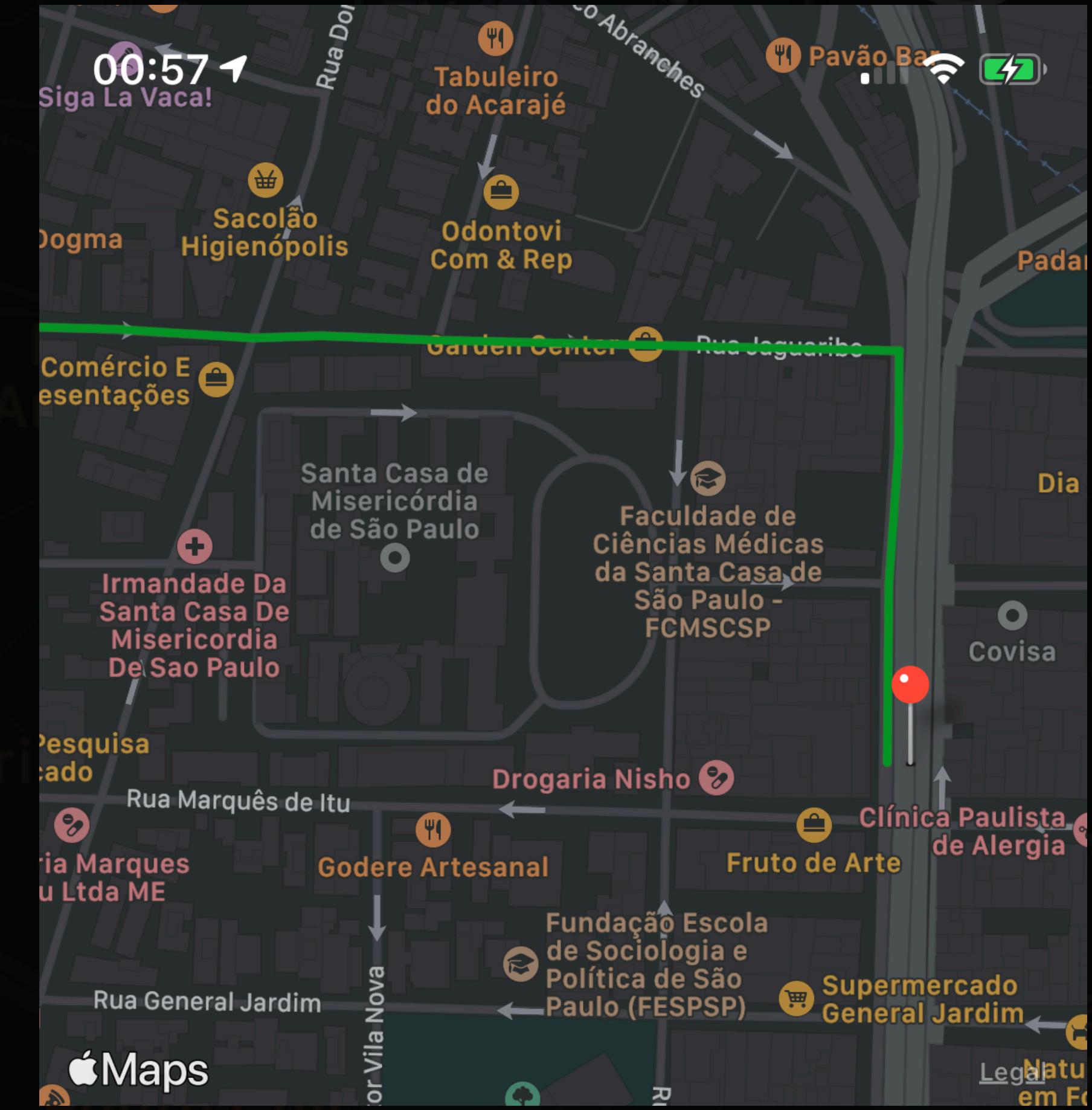
Exemplo

```
override func touchesEnded(_ touches: Set<UITouch>, with event: UIEvent?) {  
    if let touch = touches.first {  
        let position :CGPoint = touch.location(in: view)  
        let coord = map.convert(position, toCoordinateFrom: self.view)  
  
        map.addOverlays([MKCircle(center: coord, radius: CLLocationDistance(integerLiteral: 50))])  
    }  
}  
  
func mapView(_ mapView: MKMapView, rendererFor overlay: MKOverlay) -> MKOverlayRenderer {  
    if let overlay = overlay as? MKCircle {  
        let circleRenderer = MKCircleRenderer(circle: overlay)  
        circleRenderer.fillColor = UIColor.green  
        circleRenderer.alpha = 0.3  
        circleRenderer.strokeColor = UIColor.black  
        circleRenderer.lineWidth = 2  
        return circleRenderer  
    }  
  
    return MKCircleRenderer()  
}
```

Direções

MKDirectionRequest

O objeto MKDirections em conjunto com o método MKDirections.Request(), prove uma forma simples para pesquisar rotas a partir de duas coordenadas. A busca retorna uma lista com as possíveis rotas do servidor da Apple.



MKDirectionRequest

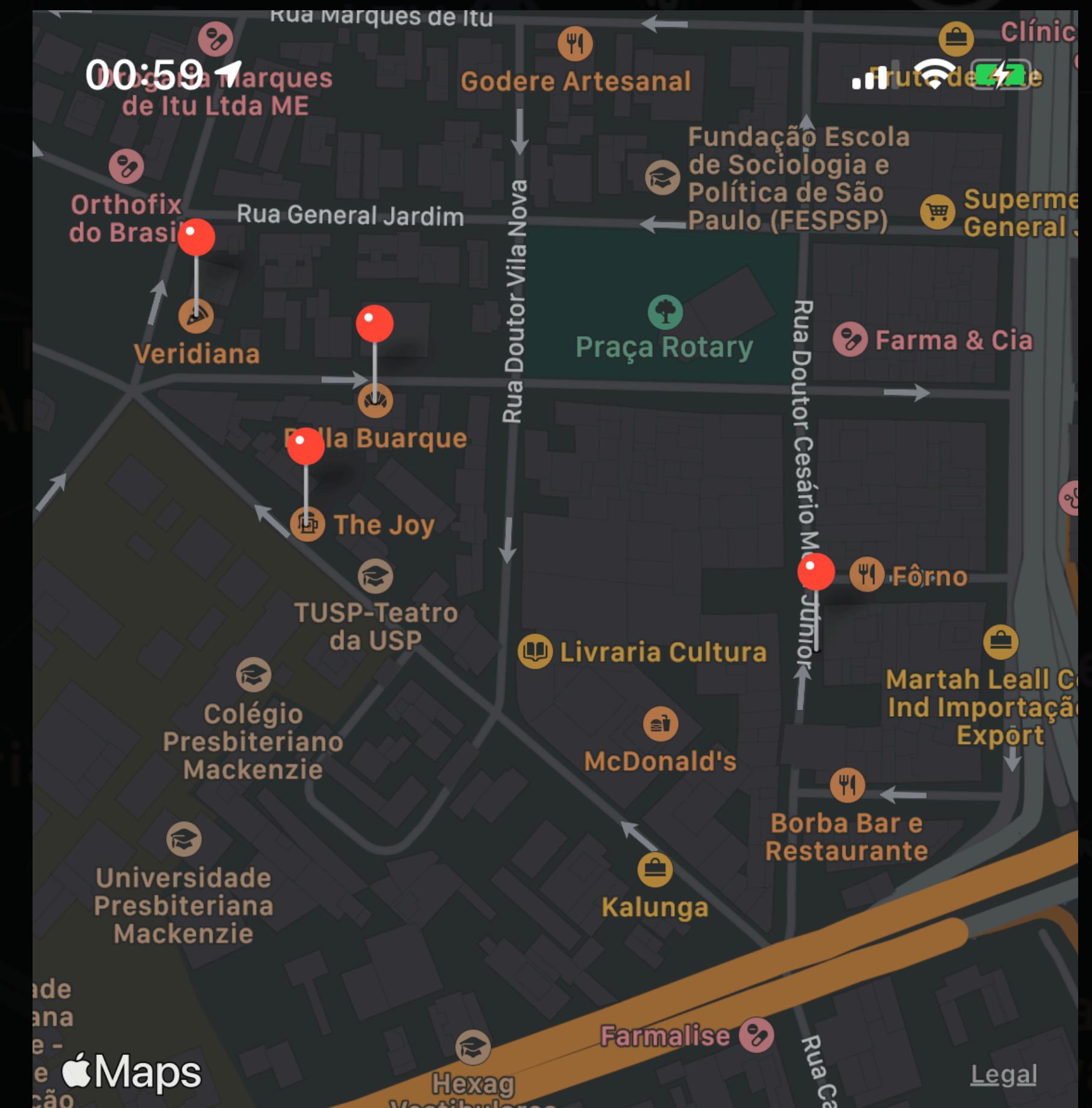
Exemplo

```
func requestDirectionsTo(source : CLLocationCoordinate2D, destination : CLLocationCoordinate2D) {  
    let request = MKDirections.Request()  
    request.source = MKMapItem(placemark: MKPlacemark(coordinate: source))  
    request.destination = MKMapItem(placemark: MKPlacemark(coordinate: destination))  
    request.requestsAlternateRoutes = true  
    request.transportType = .automobile  
  
    let directions = MKDirections(request: request)  
    directions.calculate { (response, error) in  
        if error == nil {  
            if let routes = response?.routes {  
                for route in routes {  
                    print(route.distance)  
                    self.map.addOverlays([route.polyline])  
                }  
            }  
        }  
    }  
}
```

Busca por locais

MKLocalSearch

O objeto MKLocalSearch junto com o método MKLocalSearch.Request(), prove uma forma simples para pesquisar locais em uma determinada região. A busca retorna uma lista de itens que podem ser adicionados ao mapa.



MKLocalSearchRequest

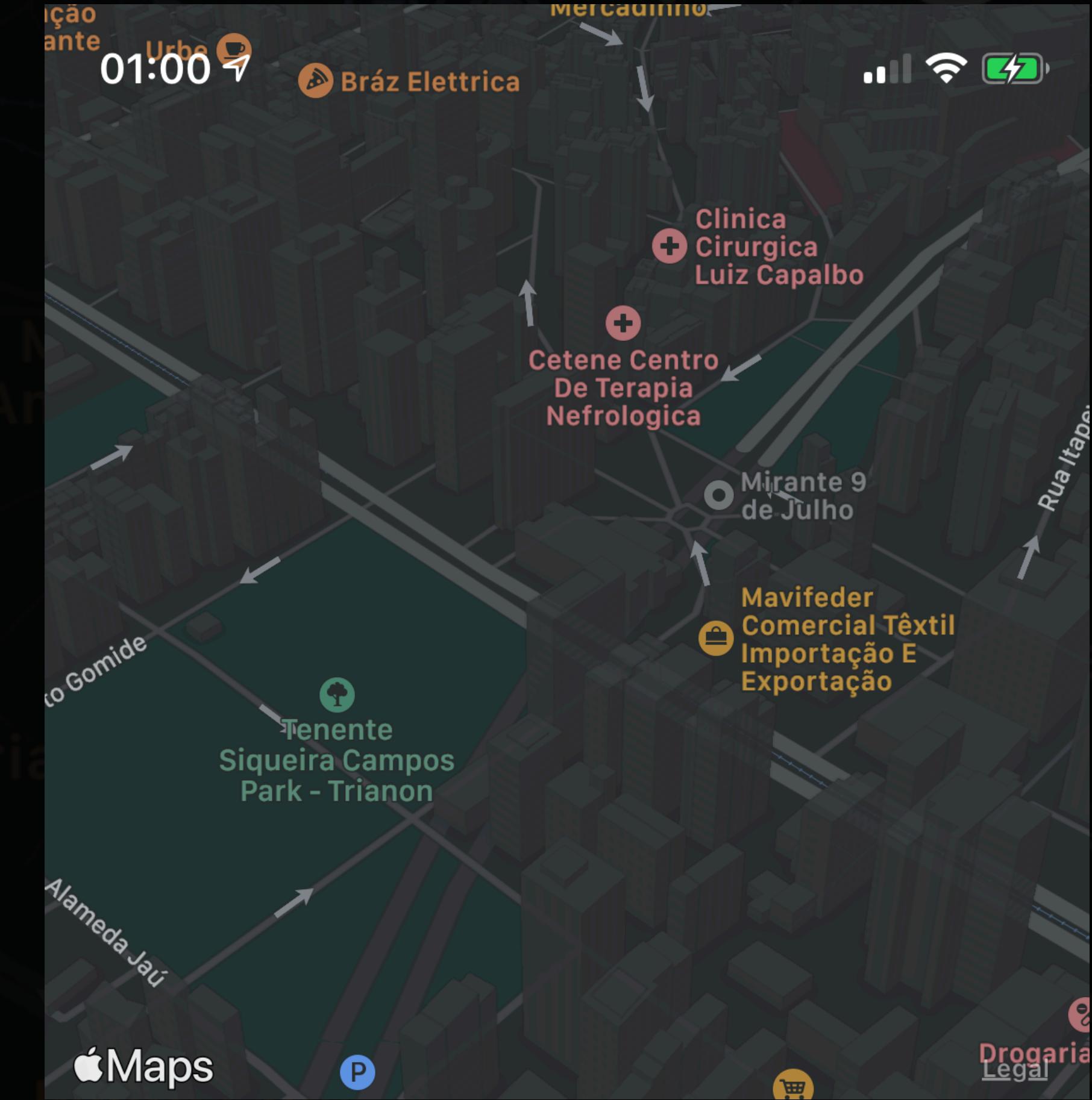
Exemplo

```
func localSearch() {  
    let request = MKLocalSearch.Request()  
    request.naturalLanguageQuery = "restaurant"  
    request.region = map.region  
  
    let search = MKLocalSearch(request: request)  
    search.start { (response, error) in  
        if error == nil {  
            if let res = response {  
                for local in res.mapItems {  
                    print(local.name)  
                    self.map.addAnnotation(local.placemark)  
                }  
            }  
        }  
    }  
}
```

Camera Virtual

MKMapCamera

O objeto MKMapCamera cria uma câmera virtual que define a aparência do mapa. Levando em conta dois pontos e uma altura, essa câmera proporciona não apenas um mapa plano, e sim uma experiência em 3D.



MKMapCamera

Exemplo

```
func mapCamera(userCoordinate: CLLocationCoordinate2D, eyeCoordinate :  
CLLocationCoordinate2D) {  
  
    let mapCamera = MKMapCamera(lookingAtCenter: userCoordinate,  
fromEyeCoordinate: eyeCoordinate, eyeAltitude: 400.0)  
  
    let annotation = MKPointAnnotation()  
    annotation.coordinate = userCoordinate  
    map.addAnnotation(annotation)  
    map.setCamera(mapCamera, animated: true)  
}
```

Resources

- MapKit - <https://developer.apple.com/reference/mapkit>
- CoreLocation - <https://developer.apple.com/reference/corelocation>
- Código de exemplo: https://github.com/joaquimp/aula_iOS-MapKit