



Materia:	Programación móvil II	
Práctica:	Regreso a casa	
Alumno(s):	Luis Alberto Ignacio Esteban	Fecha: 10/05/2020

Objetivo

Crear una aplicación que genere una ruta desde tu ubicación a un punto determinado. Utilizando los mapas de OpenStreetMap.



Temas del plan de estudios

Mapas y GPS.

API v2+ de Google maps o OSMdroid, osmdroidbonuspack

Uso de servicios GPS

Material

- Android Studio versión 3.6.3
- 1 Equipo de cómputo con un procesador a 2.0GHz y con una memoria RAM de 4gb.
- 1 Smartphone con sistema operativo Android con Lollipop o superior.
- 1 Cable USB para el Smartphone

Marco Teórico

Un mapa es una representación gráfica simplificada de un territorio con propiedades métricas sobre una superficie bidimensional que puede ser plana, esférica o incluso poliédrica. Las



propiedades métricas del mapa dependen de la proyección utilizada, y posibilitan la toma de medidas de distancias, ángulos o superficies sobre él y su relación con la realidad, en algunos casos aplicando coeficientes conocidos para la corrección de las medidas.

Como tal un mapa es una herramienta que nos ayuda a visualizar países, islas, penínsulas, continentes, países, estados, municipios, delegaciones, colonias, municipios y calles, los mapas siempre se han utilizado ya sea para conocer lugares que nunca has visitado o lugares de los que quieres tener más información de cómo llegar.

Así como la tecnología va evolucionando, esto conlleva a que las herramientas que son perceptibles al tacto se van creando de forma digital ayudando a que tengas una mejor experiencia de usuario.



Google Maps

Actualmente la mayoría de las personas utilizan algún mapa para realizar alguna actividad, uno de los mapas más populares son Google Maps, es uno de los principales mapas utilizados alrededor del mundo con una cantidad de millones de usuarios, lo cual lo convierte en una de las favoritas por las características que este implementa.

Sin embargo para el desarrollo de aplicaciones con la API de Google no es tan bueno puesto que para utilizar los servicios de la API se tiene que pagar.

También existen otras alternativas para la creación de aplicaciones con uso de mapas, como lo es OpenStreetMap.

OpenStreetMap es un proyecto colaborativo para crear mapas editables y libres. En lugar del mapa en sí, los datos generados por el proyecto se consideran su salida principal. Los mapas se crean utilizando información geográfica capturada con dispositivos GPS móviles, ortofotografías y otras fuentes libres.



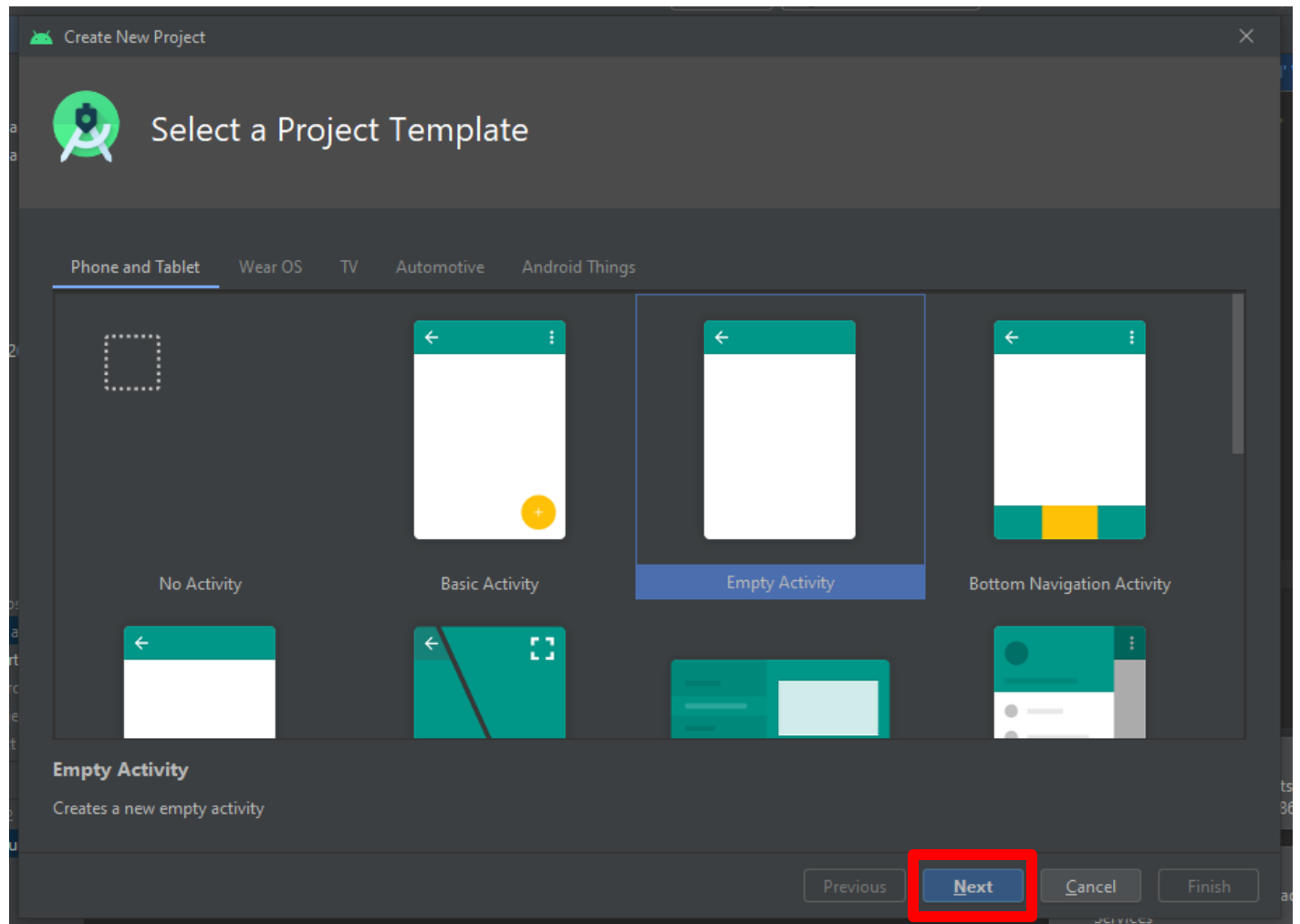
OpenStreetMap

Para el desarrollo de esta aplicación se hará uso de la librería osmdroidbonuspack versión 6.6.0, con la finalidad de no violar los derechos de autor.

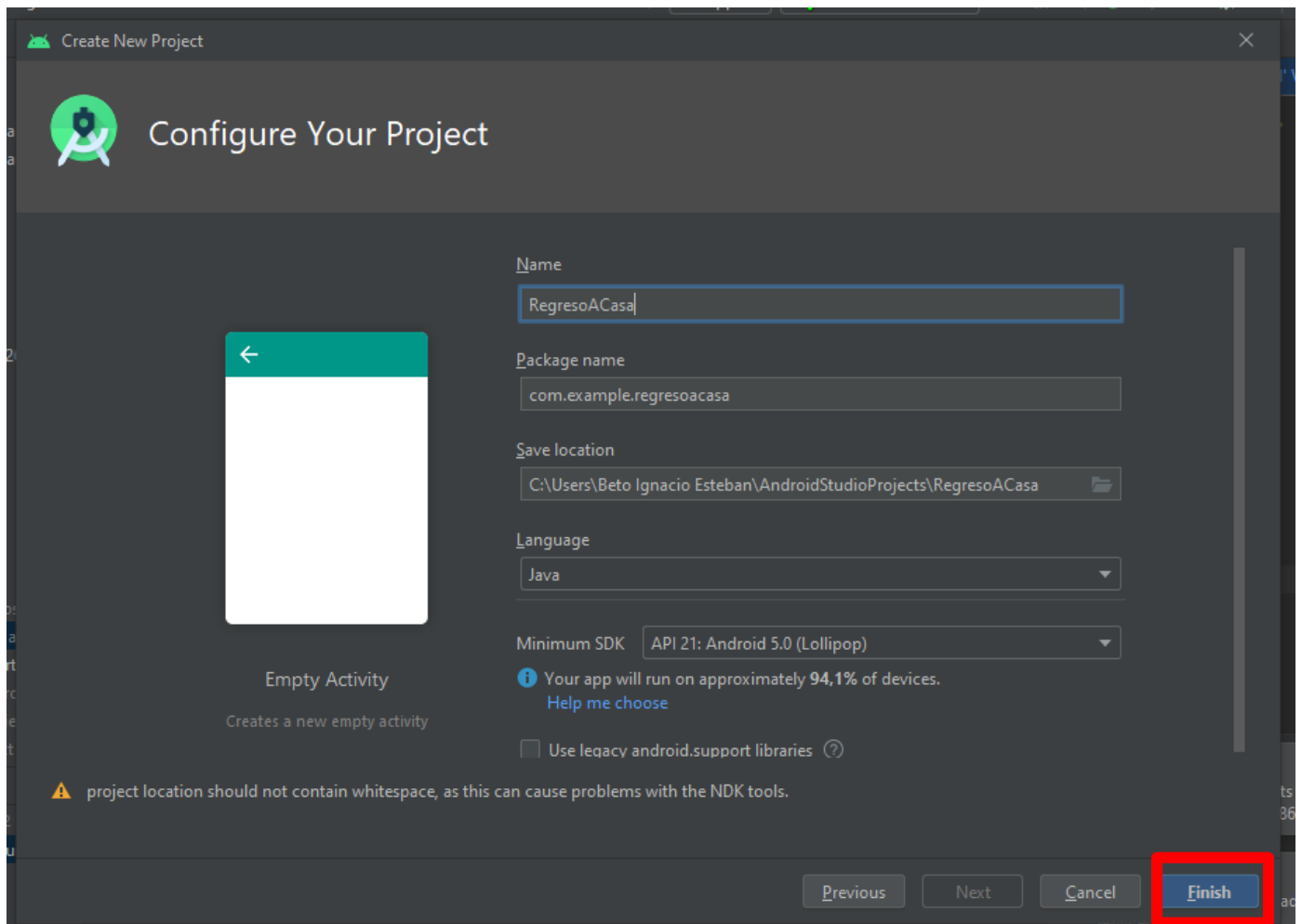


Desarrollo

Paso 1. Crear un nuevo proyecto, seleccionar un “Empty Activity”, dar clic en “Next”.



Paso 2. Asignar un nombre a la aplicación, seleccionar el lenguaje en el que se desarrollara la aplicación e indicar la versión mínima del SDK para la que estará disponible la aplicación. Dar clic en “Finish”.



Paso 3. Agregar dependencias

Dirigirse a **Gradle Scripts>build.gradle(Project: RegresoACasa)** agregar el repositorio jitpack
“maven{url “https://jitpack.io” }” para utilizar la librería osmbonuspack.

```
allprojects {
    repositories {
        google()
        jcenter()
        maven { url "https://jitpack.io" }
    }
}
```

Dirigirse a **Gradle Scripts>build.gradle(Module: app)**, en el apartado de dependencias agregar
“com.github.MKergall:osmbonuspack:6.6.0” y “com.google.android.gms:play-services-
location:17.0.0” para utilizar la librería osmbonuspack y los servicios de location. Después “Sync
now” para agregar la librerías.



```
implementation 'com.github.MKergall:osmbonuspack:6.6.0'  
implementation 'com.google.android.gms:play-services-location:17.0.0'
```

Paso 4. Agregar permisos

Dirigirse al AndroidManifest agregar los permisos que se necesitaran para utilizar la aplicación.

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>  
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>  
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

Dirigirse al AndroidManifest>application agregar “usesCleartextTraffic” con el fin de permitir las peticiones HTTP, esto es para realizar peticiones correctas.

Desde Android 9 Pie ahora, las solicitudes sin cifrado nunca funcionarán. Y de manera predeterminada, el Sistema esperará que use TLS de manera predeterminada. Por lo tanto, si solo realiza solicitudes a través de HTTPS.

```
<application  
    . . .  
    android:usesCleartextTraffic="true"  
  
>
```

Paso 5. Diseño de la parte grafica

Se utilizara la siguiente vista en el activity_main.xml, se utilizara una vista de mapa de OpenStretMap.

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent">  
  
    <org.osmdroid.views.MapView  
        android:id="@+id/map"  
        android:layout_width="fill_parent"  
        android:layout_height="fill_parent" />  
</LinearLayout>
```

Paso 6. Diseño de la parte lógica

Instanciar los siguientes objetos



```
private final int REQUEST_PERMISSIONS_REQUEST_CODE = 1;
private static final int REQUEST_CHECK_SETTINGS = 1001;
private FusedLocationProviderClient fusedLocationClient;
private LocationRequest locationRequest;
LocationSettingsRequest.Builder builder;
Location mCurrentLocation ;

//Hubicacion
GeoPoint myLocation, ITSUR;
private static double Latitud=20.123451,Longitud=-101.174230;
```

Desactivar la política StrictMode en onCreate, de esta manera:

```
StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);
```

Configurar el valor del usuario en onCreate

```
Configuration.getInstance().setUserAgentValue("OBP_Tuto/1.0");
```

En el método `onCreate()` de tu actividad, crea una instancia de cliente de proveedor de ubicación combinada, como se muestra en el siguiente fragmento de código.

```
fusedLocationClient = LocationServices.getFusedLocationProviderClient(this);
```

Agregar los métodos que se utilizarán para la validación de los permisos en tiempo de ejecución

```
@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
    ArrayList<String> permissionsToRequest = new ArrayList<>();
    for (int i = 0; i < grantResults.length; i++) {
        permissionsToRequest.add(permissions[i]);
    }
    if (permissionsToRequest.size() > 0) {
        ActivityCompat.requestPermissions(
            this,
            permissionsToRequest.toArray(new String[0]),
            REQUEST_PERMISSIONS_REQUEST_CODE);
    }
}

private void requestPermissionsIfNecessary(String[] permissions) {
    ArrayList<String> permissionsToRequest = new ArrayList<>();
    for (String permission : permissions) {
        if (ContextCompat.checkSelfPermission(this, permission)
```



```
        != PackageManager.PERMISSION_GRANTED) {  
            // Permission is not granted  
            permissionsToRequest.add(permission);  
        }  
    }  
    if (permissionsToRequest.size() > 0) {  
        ActivityCompat.requestPermissions(  
            this,  
            permissionsToRequest.toArray(new String[0]),  
            REQUEST_PERMISSIONS_REQUEST_CODE);  
    }  
}
```

Validar la versión del SDK y pedir permisos declarados en Manifest

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){  
    requestPermissionsIfNecessary(new String[] {  
        Manifest.permission.WRITE_EXTERNAL_STORAGE,  
        Manifest.permission.ACCESS_NETWORK_STATE,  
        Manifest.permission.INTERNET,  
        Manifest.permission.ACCESS_COARSE_LOCATION,  
        Manifest.permission.ACCESS_FINE_LOCATION,  
        Manifest.permission.ACCESS_WIFI_STATE  
    });  
}
```

Crear un objeto de tipo contexto

```
Context ctx = getApplicationContext();  
//Punto que se usaran como coordenadas  
ITSUR = new GeoPoint(20.139464, -101.150731); //Se le asigna las coordenadas del ITSUR  
myLocation = new GeoPoint(20.123456, -101.174243);
```

Configuración del mapView que se mostrara en el activity_main.xml

```
//Configuración del mapa  
MapView map = (MapView) findViewById(R.id.map);  
map.setTileSource(TileSourceFactory.MAPNIK);  
//TileSourceFactory.USGS_SAT vista satélite  
//osmdroid ofrece diferentes tipos de vistas de mapas  
//Habilitar los botones de Zoom en el mapa  
map.setBuiltInZoomControls(true);  
map.setTilesScaledToDpi(true);  
  
//Centrar el mapa con la coordenadas del Geopint ITSUR  
map.getController().setCenter(ITSUR);  
//Realizar un zoom de 9  
map.getController().setZoom(9);
```



```
//Habilitar el zoom con los dedos
map.setMultiTouchControls(true);

//Habilitar la rotacion del mapa
RotationGestureOverlay rotarMapa = new RotationGestureOverlay(ctx, map);
rotarMapa.setEnabled(true);
map.setMultiTouchControls(true);
map.getOverlays().add(rotarMapa);

//Marcas
GpsMyLocationProvider gpsMyLocationProvider = new GpsMyLocationProvider(ctx);
gpsMyLocationProvider.addLocationSource(LocationManager.NETWORK_PROVIDER);
CompassOverlay myCompassOverlay = new CompassOverlay(ctx,
    new InternalCompassOrientationProvider(ctx), map);
myCompassOverlay.enableCompass();
map.getOverlays().add(myCompassOverlay);

MyLocationNewOverlay locationNewOverlay = new MyLocationNewOverlay(gpsMyLocationProvider, map);
locationNewOverlay.enableMyLocation();
map.getOverlays().add(locationNewOverlay);
map.invalidate();
```

Una vez que hayas creado el cliente de servicios de ubicación, podrás obtener la ubicación más reciente del dispositivo de un usuario. Cuando se conecte tu app, podrás usar el método `getLastLocation()` del proveedor de ubicación combinada para obtener la ubicación del dispositivo. La precisión de la ubicación que muestra esta llamada es determinada por la configuración de permisos que estableciste en el manifiesto de tu app.

Para solicitar la ubicación más reciente, llama al método `getLastLocation()`. En el siguiente fragmento de código se muestra la solicitud y un manejo simple de la respuesta:

```
/**
 * Metodo para mostrar La latitud y La longitud del usuario
 * mostrando La informacion en texto(TextView)
 */
public void mostrarUbicacion(){
    fusedLocationClient.getLastLocation()
        .addOnSuccessListener(this, new OnSuccessListener<Location>() {
            @Override
            public void onSuccess(Location location) {
                //Sabiedo que obtuvimos location, lo usamos
                mCurrentLocation=location;
                if (location != null) {
                    latitud = location.getLatitude();
                    longitud = location.getLongitude();
                }else {
                    latitud = 0;
                    longitud = 0;
                }
            }
        })
}
```




```
    }  
    });  
}
```

Instanciar el objeto `locationRequest`, con una configuración apropiado para mapear apps que muestran la ubicación en tiempo real. Intervalo de 10000. Intervalo de velocidad de respuesta de 5000. Prioridad de `PRIORITY_HIGH_ACCURACY`

```
protected void createLocationRequest() {  
    locationRequest = LocationRequest.create();  
    locationRequest.setInterval(10000); // Definir las milésimas de segundo para recibir  
    // actualización de ubicación  
    locationRequest.setFastestInterval(5000); // Milisegundos de segundos que la app puede controlar  
    // las actualizaciones de ubicación  
    locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY); // Definir la prioridad  
    // de la solicitud  
}
```

Llamar al método `createLocationRequest()` para obtener la configuración de `locationRequest`

```
createLocationRequest();  
// Crear el objeto LocationSettingsRequest.Builder con la configuración que se realizó al objeto  
// locationRequest  
builder = new LocationSettingsRequest.Builder().addLocationRequest(locationRequest);  
// Verificar que se cumpla la configuración de la ubicación actual:  
SettingsClient client = LocationServices.getSettingsClient(this);  
Task<LocationSettingsResponse> task = client.checkLocationSettings(builder.build());  
  
// Determinar si la configuración de la ubicación del usuario permite que los  
// servicios de ubicación creen una LocationRequest  
task.addOnSuccessListener(this, new OnSuccessListener<LocationSettingsResponse>() {  
    @Override  
    public void onSuccess(LocationSettingsResponse locationSettingsResponse) {  
        // All location settings are satisfied. The client can initialize  
        // location requests here.  
        // ...  
    }  
});  
  
task.addOnFailureListener(this, new OnFailureListener() {  
    @Override  
    public void onFailure(@NonNull Exception e) {  
        if (e instanceof ResolvableApiException) {  
            // Location settings are not satisfied, but this can be fixed  
            // by showing the user a dialog.  
            try {  
                // Show the dialog by calling startResolutionForResult(),  
                // and wait for the response in onActivityResult().  
            } catch (IOException ex) {  
                // Unable to resolve the API.  
            }  
        }  
    }  
});
```



```
// and check the result in onActivityResult().
ResolvableApiException resolvable = (ResolvableApiException) e;
resolvable.startResolutionForResult(MainActivity.this,
    REQUEST_CHECK_SETTINGS);
Toast.makeText(getApplicationContext(),"Sin fallas con la aplicacion",
    Toast.LENGTH_SHORT).show();
} catch (IntentSender.SendIntentException sendEx) {
    // Ignore the error.
    Toast.makeText(getApplicationContext(),"Error con la aplicacion",
    Toast.LENGTH_SHORT).show();
}
}
});
```

Llamar al método mostrarUbicacion, con la finalidad de obtener la latitud y longitud donde se encuentra el dispositivo.

```
this.mostrarUbicacion();
```

Asignar las coordenadas al GeoPoint myLocation. Crear un administrador de carreteras con el cual se trazara una ruta del punto A al punto B.

Crear un ArrayList de tipo GeoPoint este servirá como indicador de los puntos por donde se debe trazar la ruta, agregar los GeoPoints myLocation e ITSUR.

Se crea un Road y se le asigna la ruta que se trazara el RoadManager utilizando los GeoPoints que se encuentran en la lista. Agregar la ruta creada al mapa.

```
myLocation = new GeoPoint(latitud,longitud);
RoadManager roadManager = new OSRMRoadManager(ctx);
ArrayList<GeoPoint> waypoints = new ArrayList<GeoPoint>();
waypoints.add(myLocation);
waypoints.add(ITSUR);
Road road = roadManager.getRoad(waypoints);
Polyline roadOverlay = RoadManager.buildRoadOverlay(road);
map.getOverlays().add(roadOverlay);
map.invalidate();
```

Crear un icono de un humano en el mapa

```
locationNewOverlay = new MyLocationNewOverlay(gpsMyLocationProvider, map);
locationNewOverlay.enableMyLocation();
map.getOverlays().add(locationNewOverlay);
map.invalidate();
```

Crear etiquetas para los GeoPoints myLocation e ITSUR, asignando un nombre y las coordenadas donde se mostraran las etiquetas.

```
ArrayList<OverlayItem> puntos = new ArrayList<OverlayItem>();
puntos.add(new OverlayItem("ITSUR", "Uriangato , GTO", ITSUR));
puntos.add(new OverlayItem("Mi ubicacion", "", myLocation));
```



```
ItemizedIconOverlay.OnItemGestureListener<OverlayItem> tap = new
ItemizedIconOverlay.OnItemGestureListener<OverlayItem>() {
    @Override
    public boolean onItemLongPress(int arg0, OverlayItem arg1) {
        return false;
    }
    @Override
    public boolean onItemSingleTapUp(int index, OverlayItem item) {
        return true;
    }
};

ItemizedOverlayWithFocus<OverlayItem> capa = new ItemizedOverlayWithFocus<OverlayItem>(this,
puntos, tap);
capa.setFocusItemsOnTap(true);
map.getOverlays().add(capa);
map.invalidate();

Marker startMarker = new Marker(map);
startMarker.setPosition(myLocation);
startMarker.setAnchor(Marker.ANCHOR_CENTER, Marker.ANCHOR_BOTTOM);
startMarker.setTitle("Punto de inicio");
map.getOverlays().add(startMarker);
map.invalidate();
```

Resultado

Al instalar la aplicación en un dispositivo con versión 6 o superior se obtendrán los siguientes resultados.

Se pedirán los permisos en tiempo de ejecución.



RegresoACasa



Para mejorar la experiencia,
activa la ubicación del
dispositivo, que usa los
servicios de ubicación de
Google. ✓

NO, GRACIAS ACEPTAR

RegresoACasa



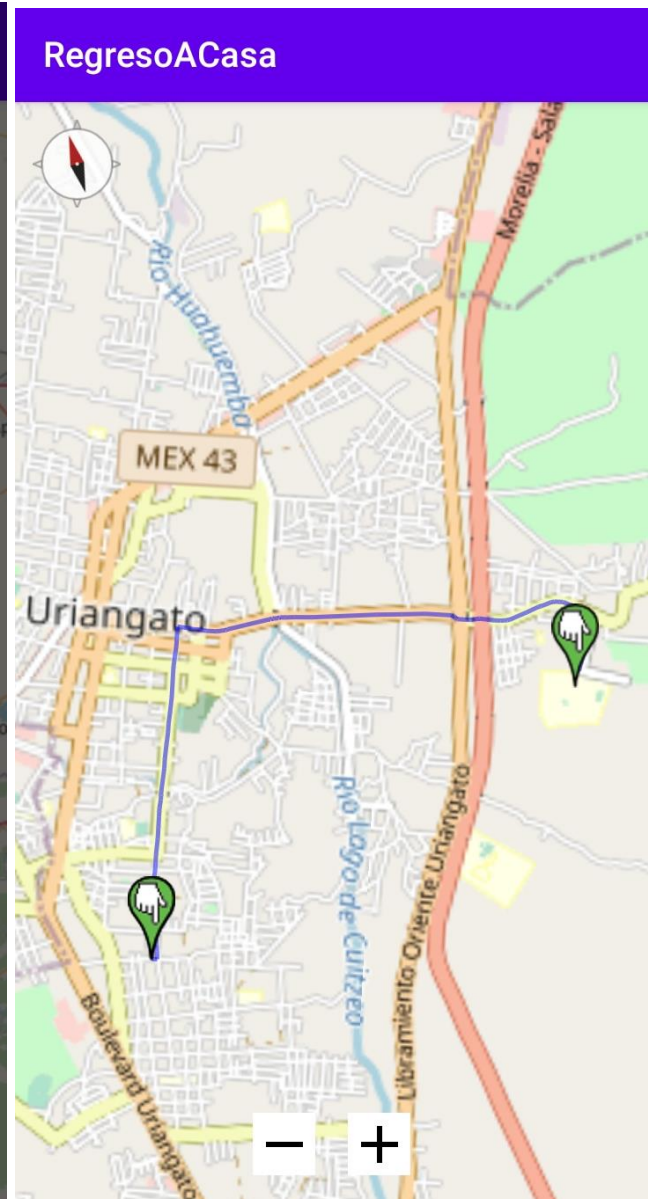
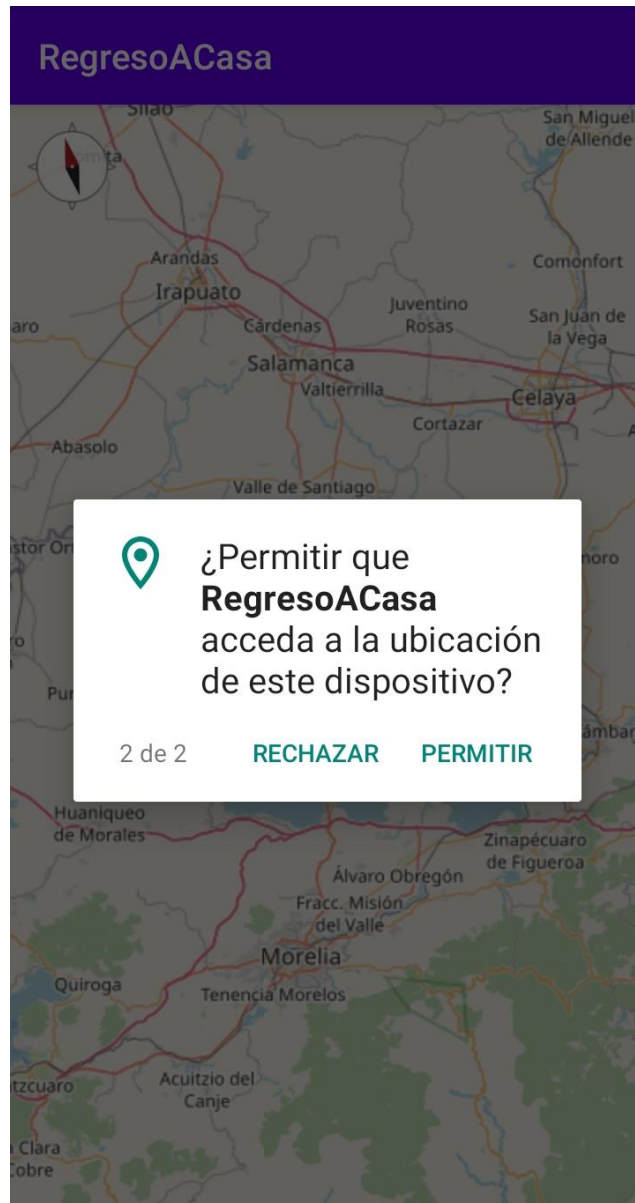
¿Permitir que
RegresoACasa
acceda a las fotos, el
contenido multimedia
y los archivos de tu
dispositivo?

1 de 2

RECHAZAR PERMITIR

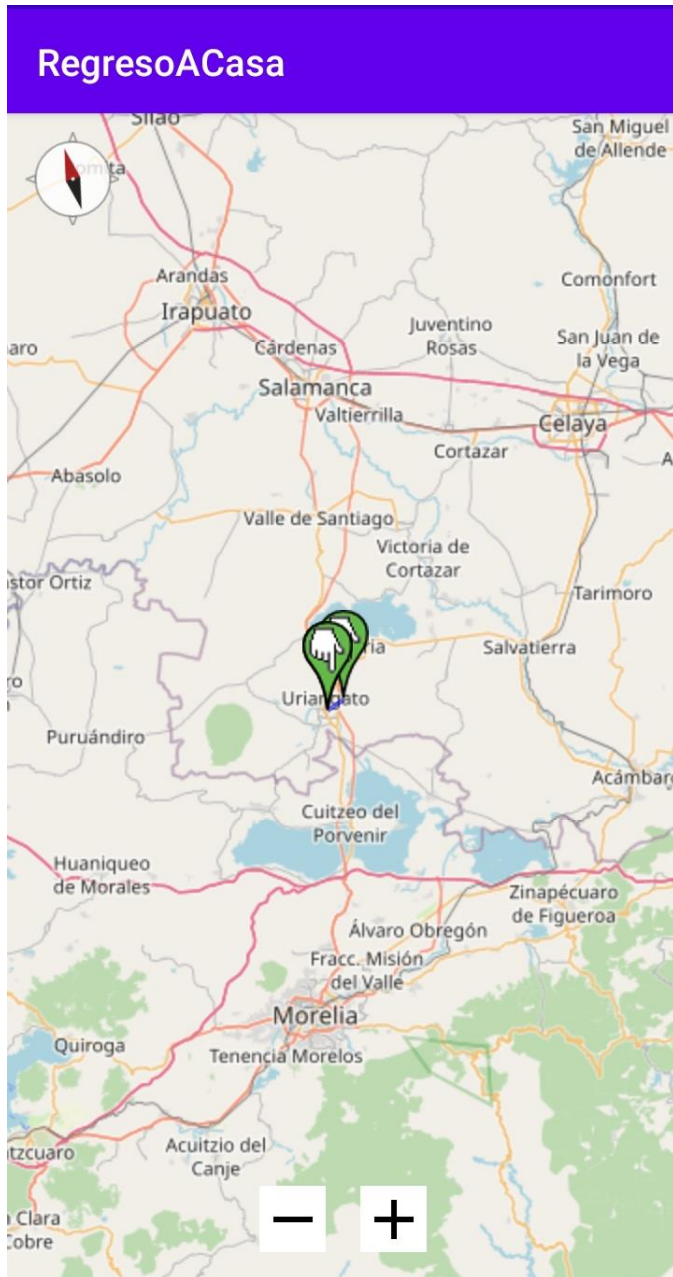


Para todos los ejemplos se utilizara como punto final la ubicación del Instituto Tecnológico Superior del Sur de Guanajuato y tomando como punto inicial la ubicación del dispositivo.



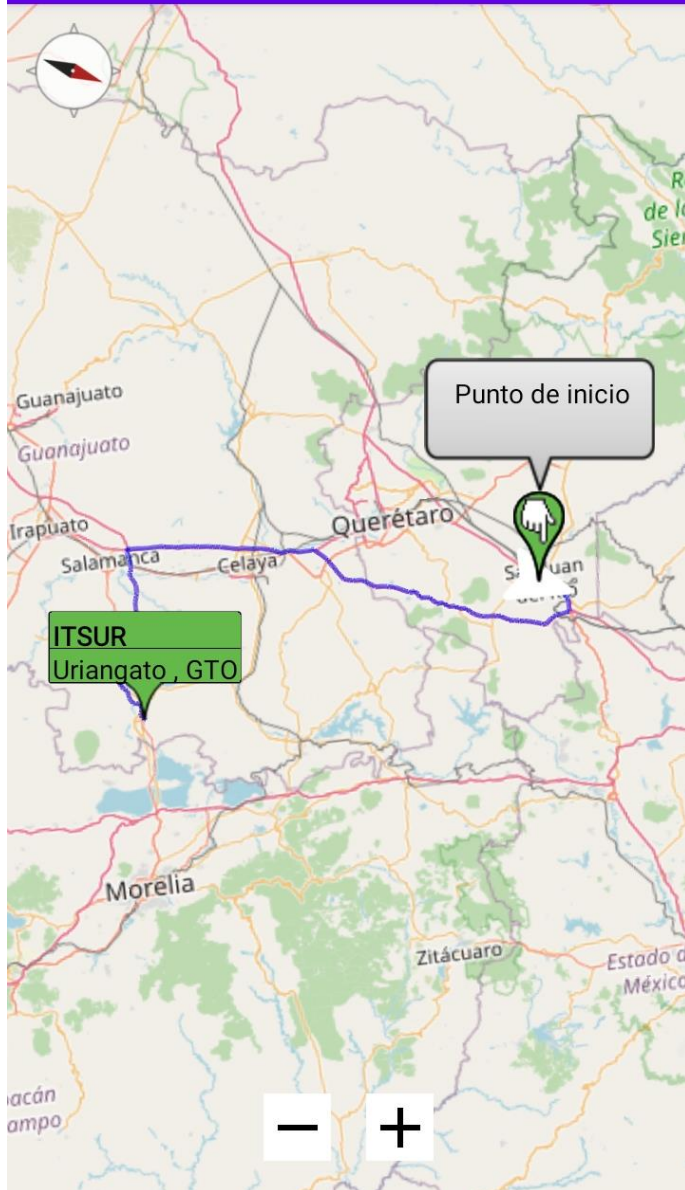


En las siguientes imágenes muestra las rutas generadas de distintos lugares.

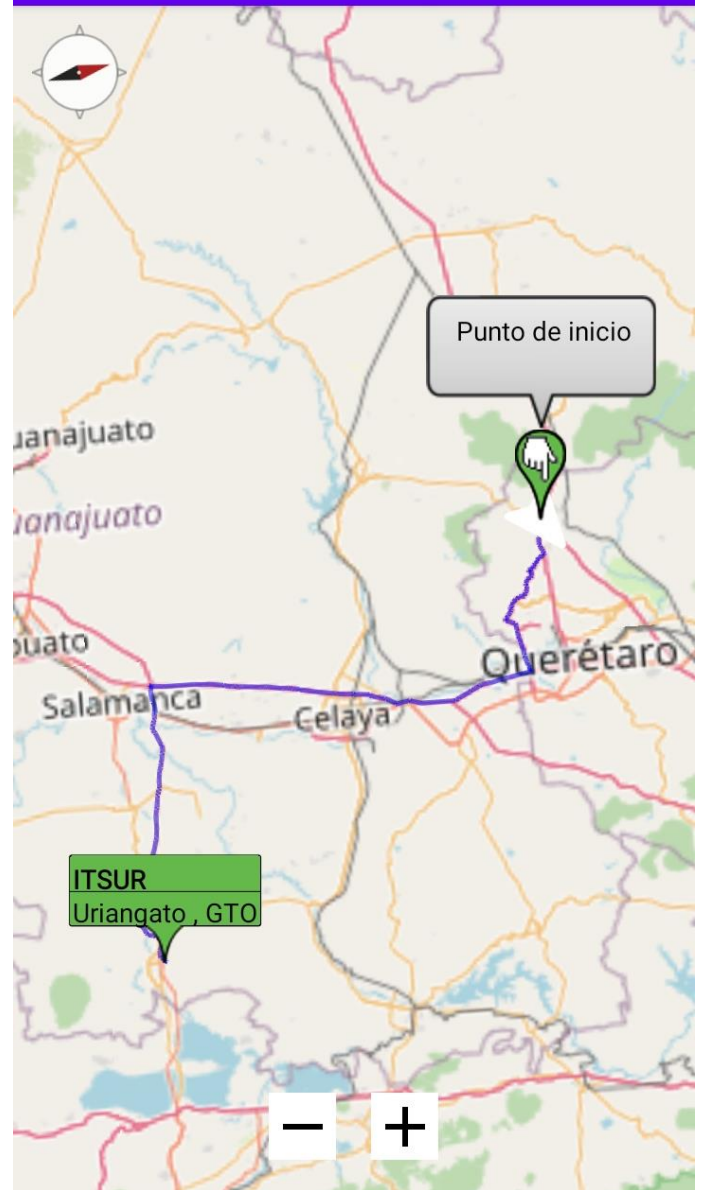




RegresoACasa

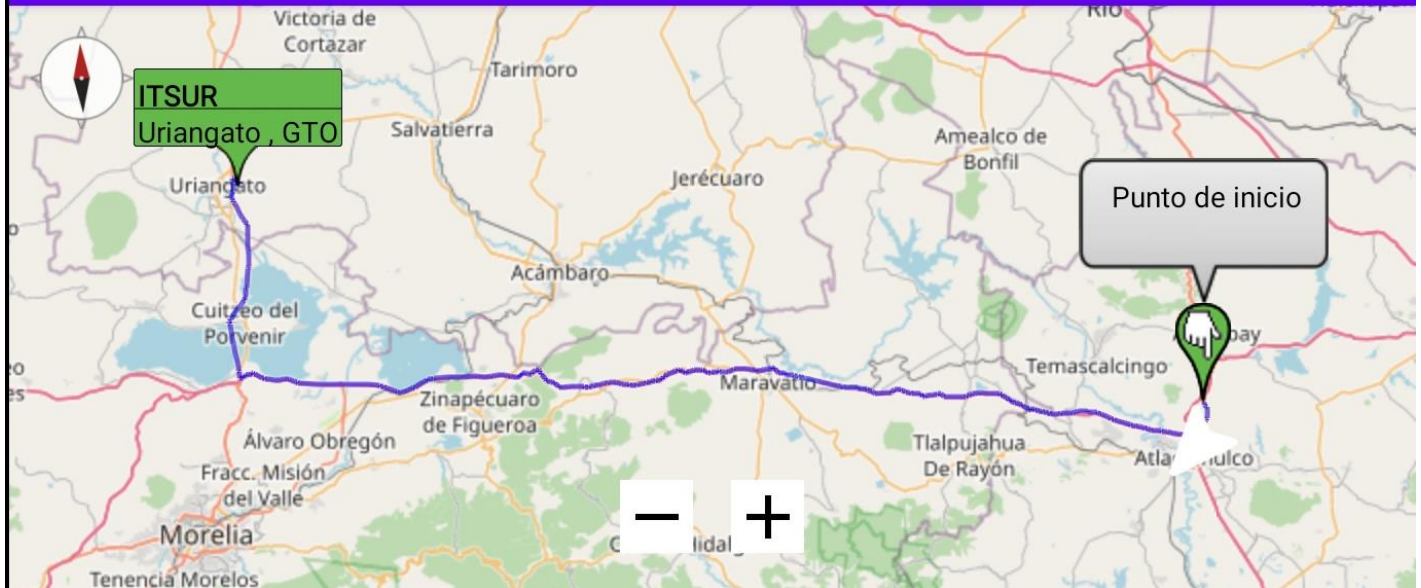


RegresoACasa





RegresoACasa



Evaluación

Conclusiones

Saber implementar mapas, ayuda a poder visualizar lugares.

Algunas de aplicaciones muy populares que utilizan mapas son Uber, DiDi Pasajero, Uber Eats, inDriver, etc. Estas apps permiten visualizar en tiempo real la ubicación ya sea de los vehículos, comida o inclusive la propia ubicación.

El uso de mapas en apps pueden ayudar a mejorar la manera en que se puede orientar las personas y realizar algún servicio.

Otras implementación son para los sistemas de rastreo de algún vehículo, esto es con el fin de poder recuperar el vehículo.

Bibliografía

<https://medium.com/@imstudio/android-8-clear-text-http-traffic-not-permitted-73c1c9e3b803>

<https://osmdroid.github.io/osmdroid/FAQ.html>

<https://www.dosmweb.com/blog/index.php?post/2016/03/28/OpenStreetMap-en-Android-Studio>



<https://github.com/osmdroid/osmdroid/wiki/How-to-use-the-osmdroid-library>

https://www.google.com/search?q=agregar+Location+overlay+osmdroid+android+studio&rlz=1C1CHBF_esMX886MX886&oq=agregar+Location+overlay+osmdroid+android+studio&aqs=chrome..69i57j33.11044j0j4&sourceid=chrome&ie=UTF-8

<https://stackoverflow.com/questions/38539637/osmbonuspack-roadmanager-networkonmainthreadexception>

Incluir librería

<https://github.com/MKergall/osmbonuspack/wiki/HowToInclude>

Problema al dibujar la ruta, solución

<https://stackoverflow.com/questions/51902629/how-to-allow-all-network-connection-types-http-and-https-in-android-9-pie>

<https://github.com/MKergall/osmbonuspack/issues/366>

<https://es.wikipedia.org/wiki/Mapa>