P.PORTO

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Programação Para Dispositivos Móveis I

GOOGLE PLAY SERVICES
LOCALIZAÇÃO, GOOGLE MAPS

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis
Ricardo Barbosa , rmb@estg.ipp.pt
Carlos Aldeias, cfpa@estg.ipp.pt

Adaptação do conteúdo dos slides de João Ramos jrmr@estg.ipp.pt e Fábio Silva fas@estg.ipp.pt

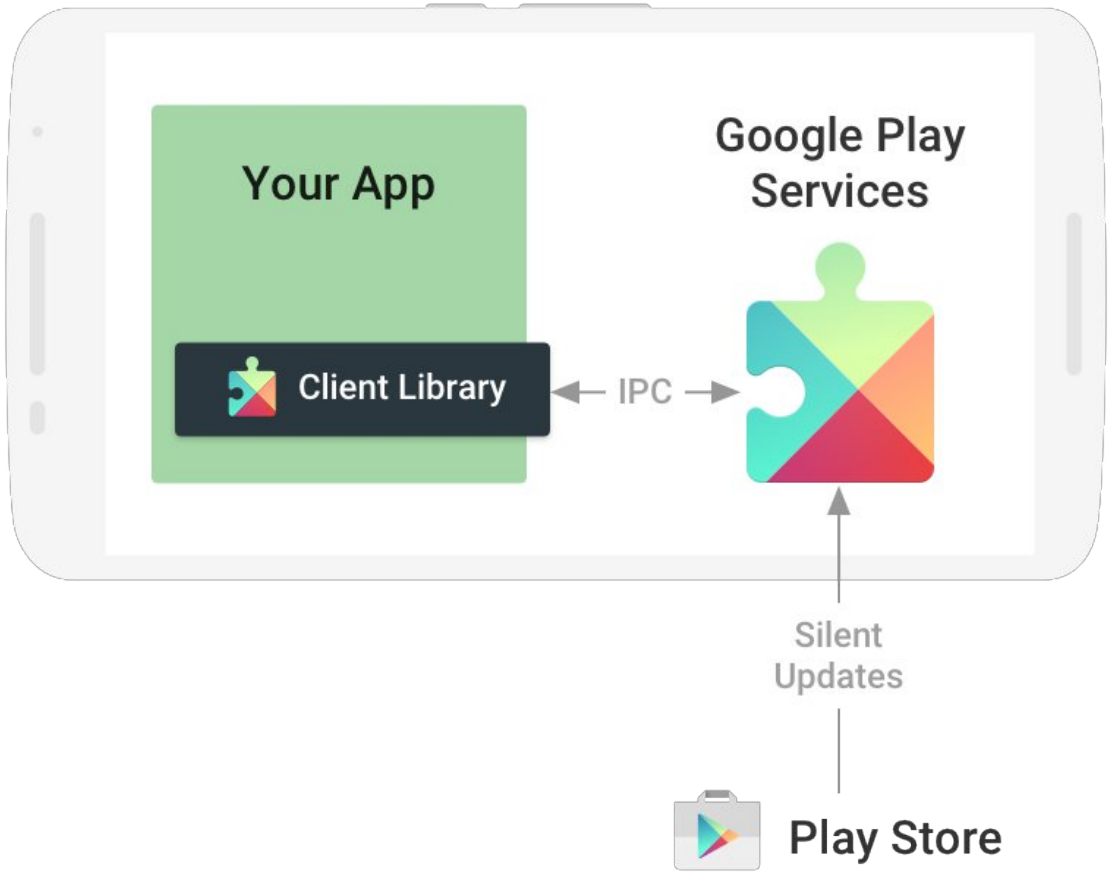
NORTE 2030 Programa Regional do Norte
PORTUGAL 2030
Cofinanciado pela União Europeia

Índice

- Google Play Services;
- Localização;
- Google Maps;
- Leitura Adicional.

Google Play Services

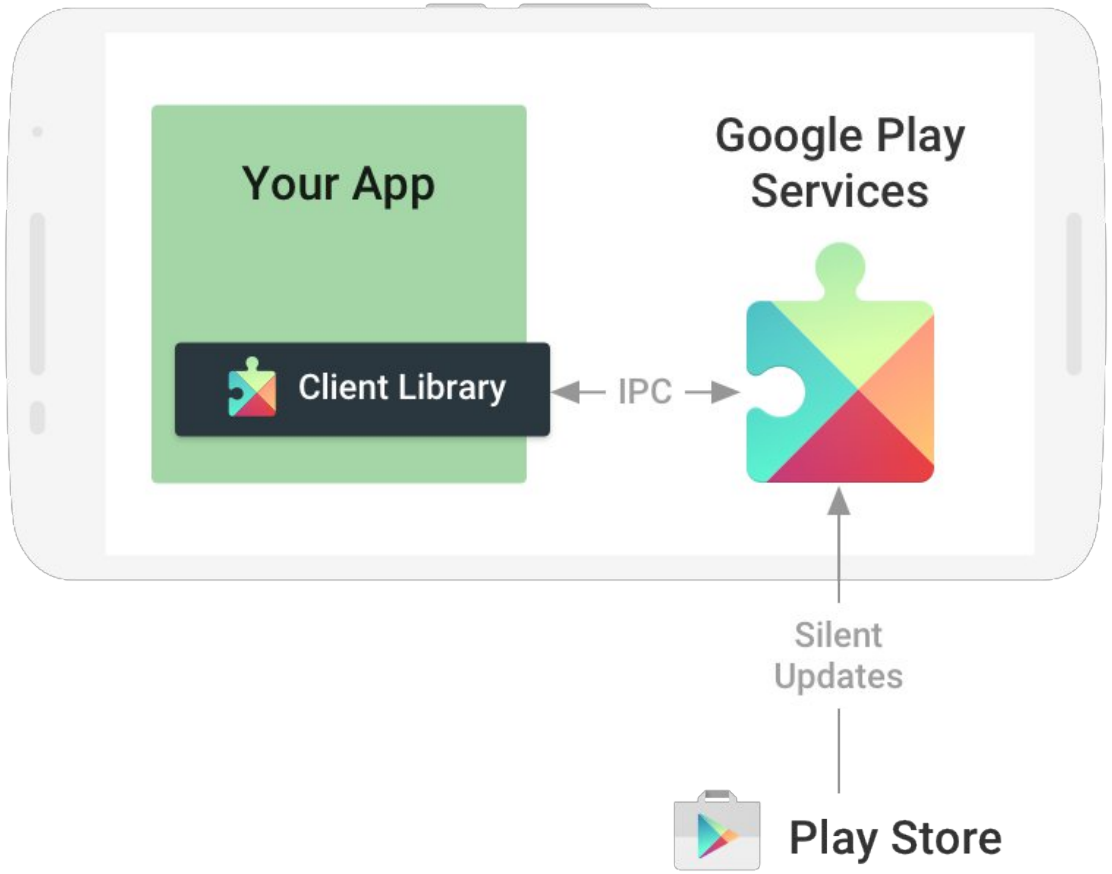
Biblioteca fornecida pela Google que permite interagir com diferentes tipos de funcionalidades, entre elas: mapas, localização, autenticação, etc.



- Os Google Play Services estão sempre a correr em background no dispositivo.

Google Play Services

- Múltiplas apps interagem ao mesmo tempo com os Google Play Services;
- Biblioteca atualizada automaticamente através da Google Play Store;
- Permite aos programadores usarem as APIs mais recentes do Google.



Google Play Services

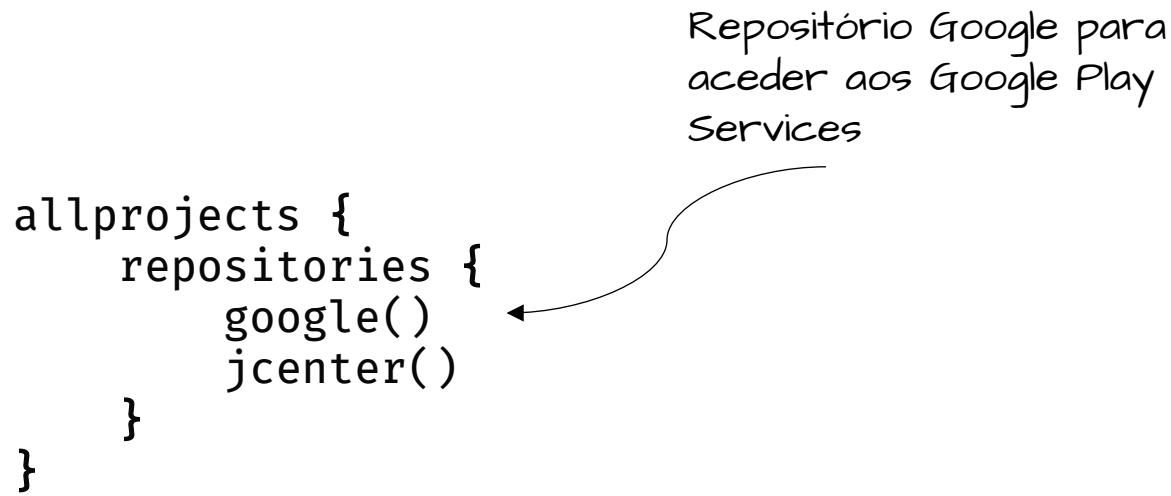
Funcionalidades

Versões atuais a abril de 2025

API	Description in build.gradle
Google Account Login	`com.google.android.gms:play-services-auth:21.3.0`
Google Actions, Base Client Library	`com.google.android.gms:play-services-base:18.7.0`
Google Awareness	`com.google.android.gms:play-services-awareness:19.0.1`
Google Cast	`com.google.android.gms:play-services-cast:22.0.0`
Cloud Messaging	`com.google.firebase:firebase-messaging:24.1.1`
Google Drive	`com.google.android.gms:play-services-drive:17.0.0`
Google Fit	`com.google.android.gms:play-services-fitness:21.2.0`
Google Location and Activity Recognition	`com.google.android.gms:play-services-location:21.3.0`
Google Mobile Ads	`com.google.android.gms:play-services-ads:24.2.0`
Mobile Vision	`com.google.android.gms:play-services-vision:20.1.3`
Google Nearby	`com.google.android.gms:play-services-nearby:19.3.0`
Google Play Game services	`com.google.android.gms:play-services-games:23.2.0`
SafetyNet	`com.google.android.gms:play-services-safetynet:18.0.1`
Google Pay	`com.google.android.gms:play-services-wallet:19.4.0`
Wear OS by Google	`com.google.android.gms:play-services-wearable:19.0.0`

Google Play Services

Configuração [build.gradle (Project)]



Google Play Services

Dependências [build.gradle (Module)]

```
dependencies {

    implementation 'com.google.android.gms:play-services-maps:18.2.0'
    implementation 'com.google.android.gms:play-services-location:21.2.0'

}
```

Dependência para Google Maps

Dependência para Serviços de Localização

Localização

Existem diferentes formas de obter a localização de um dispositivo

Android:

- apenas a ultima localização conhecida;
- atualizações periódicas da localização (x em x minutos);
- atualizações com base na distancia percorrida (x km);
- geofencing (dispositivo entrou num raio pré-definido).

Localização

O sistema operativo fornece diferentes tipos de *providers* de localização, isto é, a utilização de GPS, Wifi ou outro tipo de hardware vai variar consoante os requisitos da aplicação:

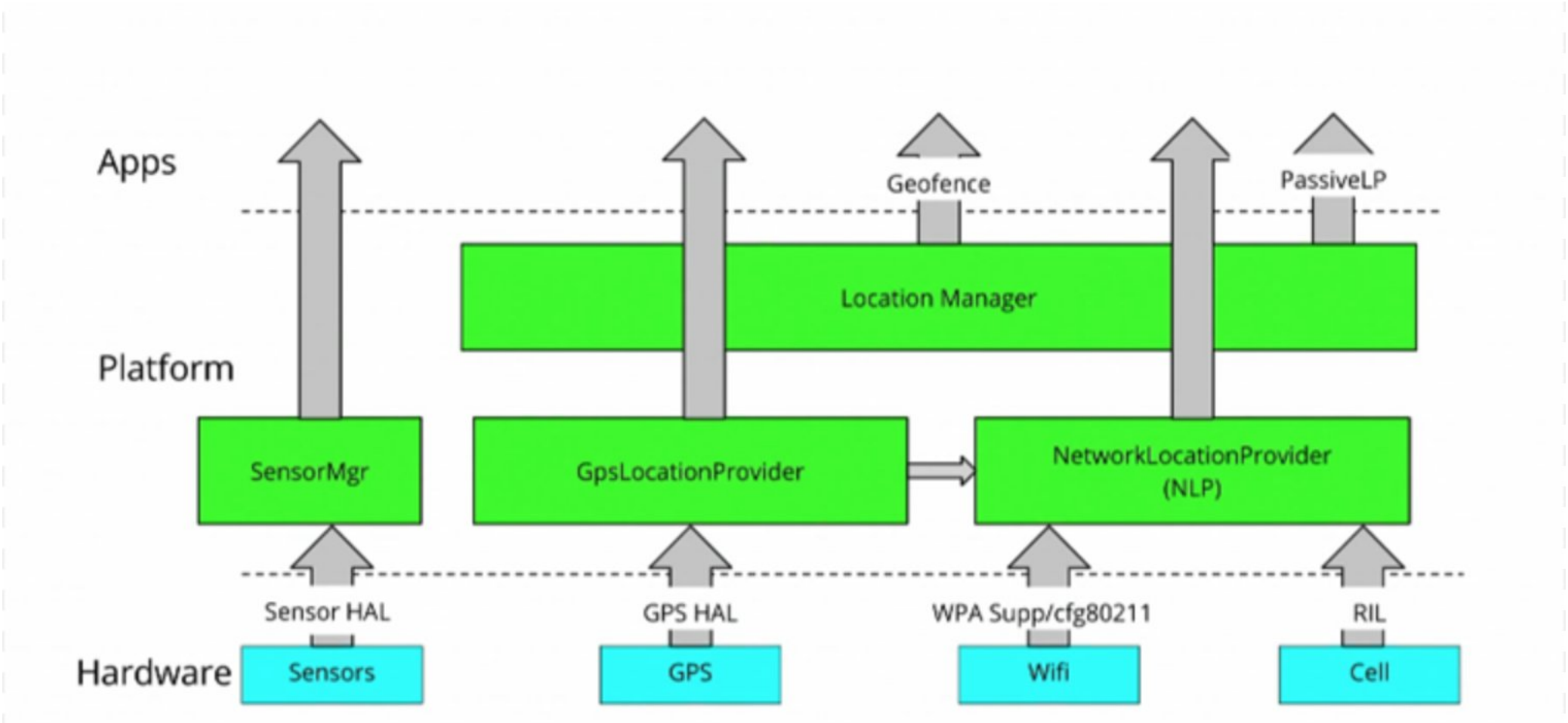
- precisão da localização (accuracy);
- consumo de bateria;
- periodicidade das atualizações à localização (30 em 30 segundos, hora em hora).

Localização

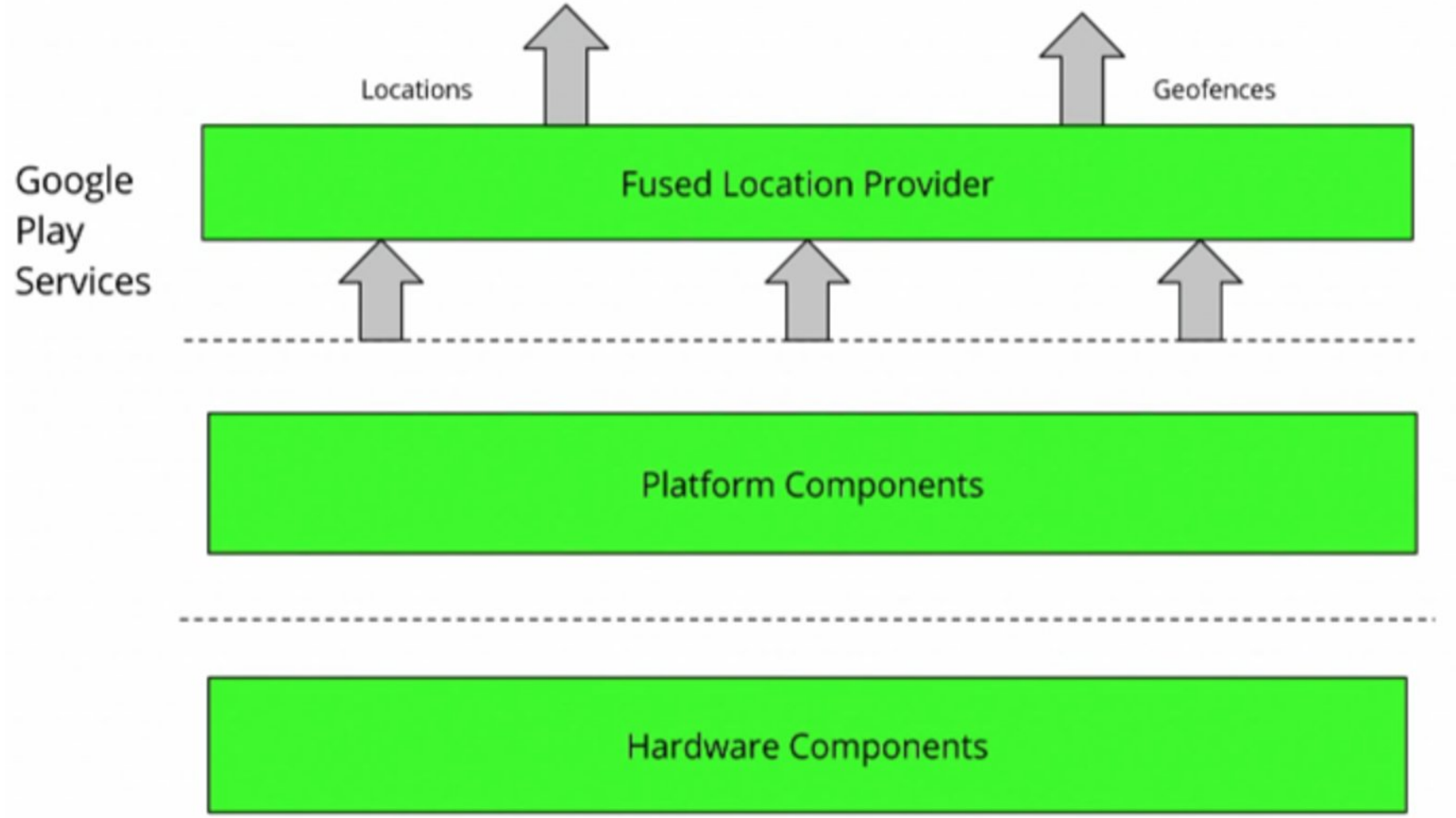
Providers de localização

	GPS	WiFi	Cell	Sensors
Power				
Accuracy				
Coverage				

Localização
API Nativa de Localização



Localização
API Google Play Services de Localização



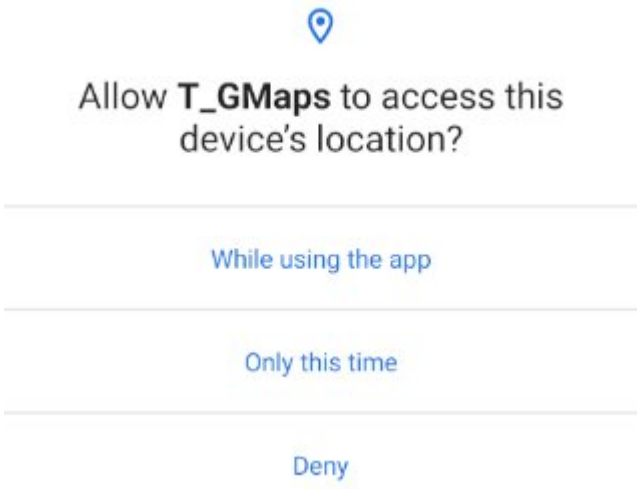
Localização

Permissões [AndroidManifest.xml]


```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Localização

Permissões [MainActivity.java]



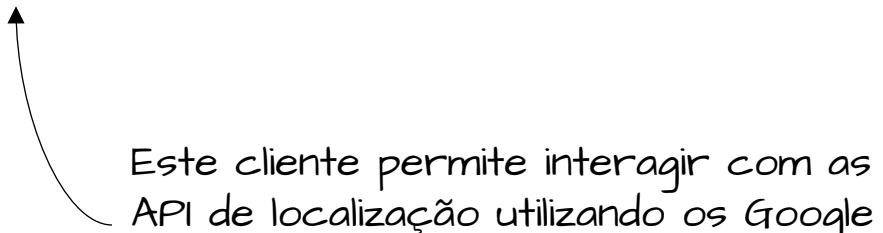

```
private static final int REQUEST_FINE_LOCATION = 100;

@Override
protected void onCreate(Bundle savedInstanceState) {
    (…)
    getLastLocation();

}

private void getLastLocation(){
    if(ActivityCompat.checkSelfPermission(this,
            Manifest.permission.ACCESS_FINE_LOCATION) ≠ PackageManager.PERMISSION_GRANTED){
        requestPermissions();
        return;
    }
}

private void requestPermissions(){
    ActivityCompat.requestPermissions(this,
            new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
            REQUEST_FINE_LOCATION);
}
```

Nas versões mais recentes de Android temos de verificar se o utilizador deu permissões à aplicação para aceder aos recursos de localização

Localização

Cliente de localização [MainActivity.java]

```
private FusedLocationProviderClient fusedLocationProviderClient;


@Override
protected void onCreate(Bundle savedInstanceState) {
    (…)

    fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this);
}
```



Localização

Obter última localização [MainActivity.java]

```
private void getLastLocation(){

    if(ActivityCompat.checkSelfPermission(this,
            Manifest.permission.ACCESS_FINE_LOCATION) ≠ PackageManager.PERMISSION_GRANTED){
        requestPermissions();
        return;
    }
    fusedLocationProviderClient.getLastLocation()
        .addOnSuccessListener(this,
            new OnSuccessListener<Location>() {
                @Override
                public void onSuccess(Location location) {
                    if(location ≠ null){
                        //TODO: Código em caso de sucesso
                    }
                }
            })
        .addOnFailureListener(this, new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                //TODO: Código em caso de erro
            }
        });
}
```

Para obter a última localização conhecida do dispositivo devemos invocar o cliente fused e adicionar o listener de sucesso e erro.

Localização
Atualizações periódicas

Para obter atualizações periódicas de localização é necessário criar um **LocationRequest** e um **LocationCallback**.

- **LocationRequest** define os parâmetros da localização que queremos obter (precisão, periodicidade, timeout);
- **LocationCallback** é o *listener* que vai receber periodicamente uma lista de localizações.

Localização

Atualizações periódicas (LocationRequest) [MainActivity.java]

```
private LocationRequest locationRequest;


@Override
protected void onCreate(Bundle savedInstanceState) {
    (…)

    locationRequest = new LocationRequest();
    locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    locationRequest.setInterval(10000);
    locationRequest.setFastestInterval(5000);
}
```

Localização

Atualizações periódicas (LocationCallback) [MainActivity.java]

```
private LocationCallback locationCallback;

@Override
protected void onCreate(Bundle savedInstanceState) {

    (…)

    locationCallback = new LocationCallback(){
        @Override
        public void onLocationResult(LocationResult locationResult) {
            super.onLocationResult(locationResult);
            for(Location location : locationResult.getLocations()){
                //TODO: Atualizar UI com a localização
            }
        }
    };
}
```

Localização

Iniciar Atualizações periódicas[MainActivity.java]

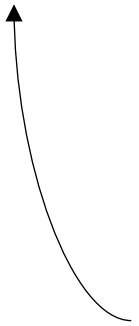
```
private void startLocationUpdates(){
    //TODO: Verificar permissões

    (…)

    fusedLocationProviderClient.requestLocationUpdates(
            locationRequest,
            locationCallback,
            null);
}
```

Para iniciar as atualizações periódicas de localização, devemos fazer um request ao cliente

Localização

Parar Atualizações periódicas[MainActivity.java]

```
private void stopLocationUpdates(){

    fusedLocationProviderClient.removeLocationUpdates(locationCallback);

}
```



Para parar as atualizações periódicas de localização, basta remover o LocationCallback do cliente

Mapas

As aplicações em Android podem usar mapas e Informação geográfica utilizando bibliotecas externas:

- Google Maps;
- Here Maps;
- Leaflet.

Para efeitos de exemplo iremos utilizar o Google Maps presente nos Play Services vistos anteriormente.

Google Maps
API Key

É necessário obter uma API key para aceder aos serviços do Google Maps.

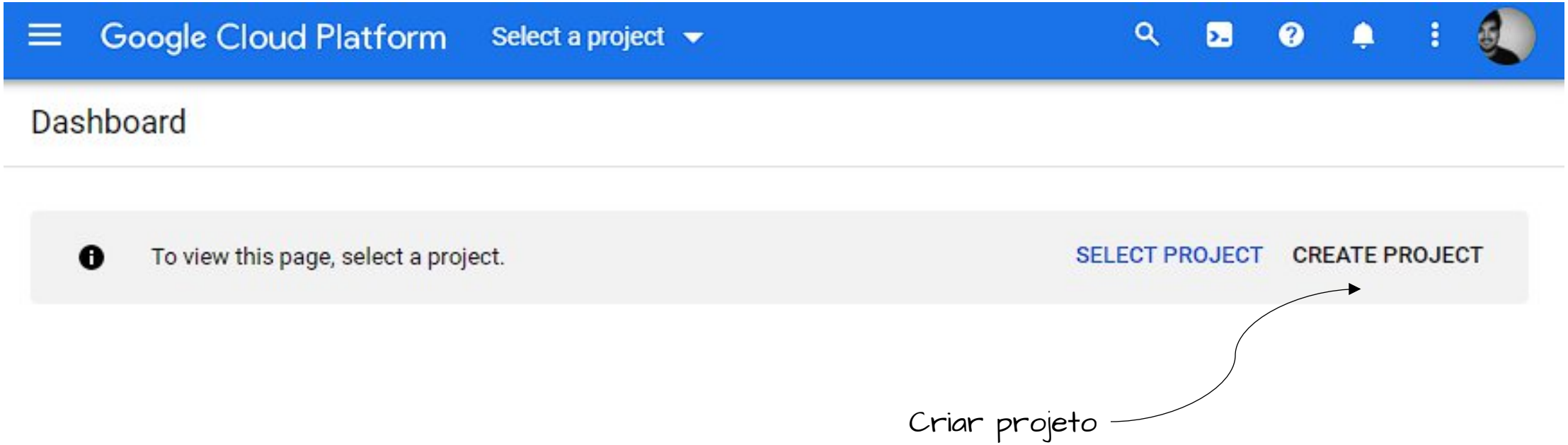
- A chave pode ser obtida no seguinte endereço:

  https://developers.google.com/maps/documentation/android-sdk/get-api-key

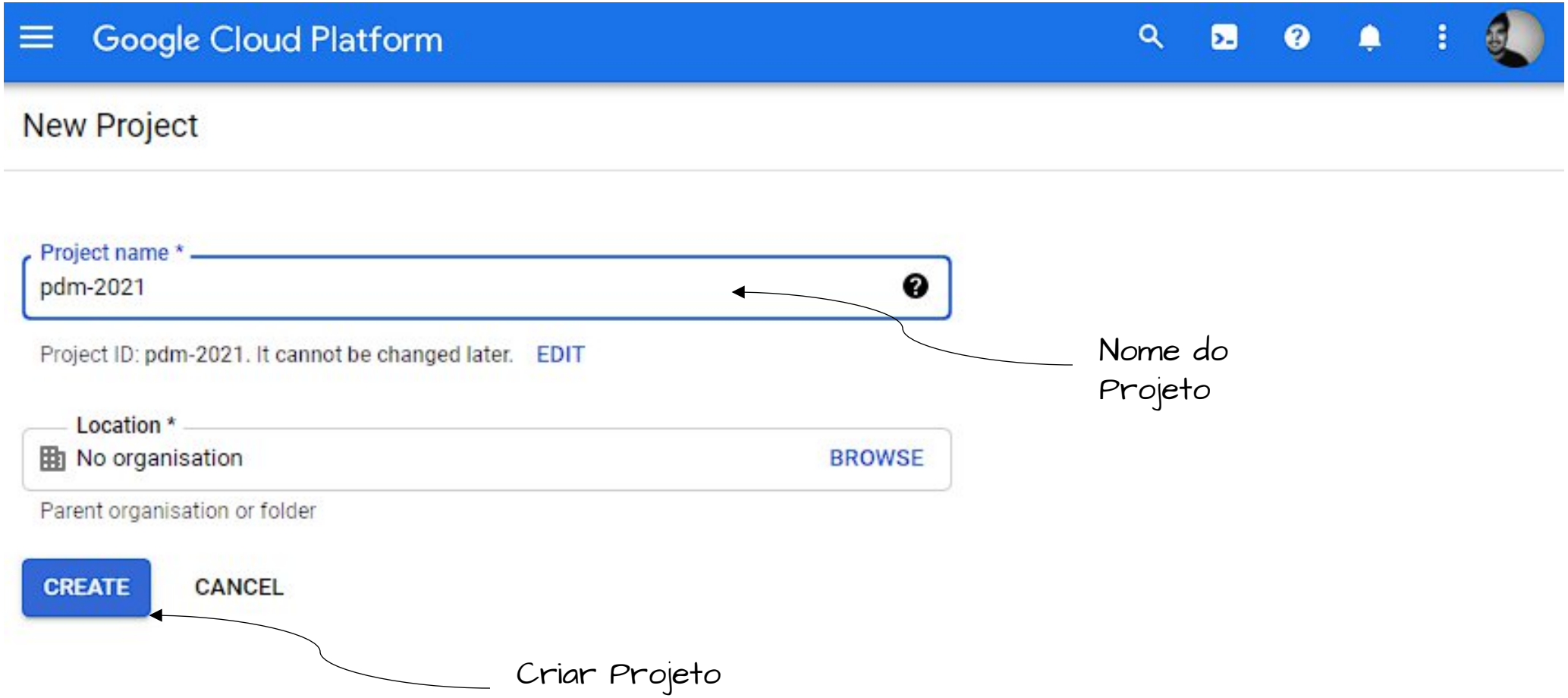
Google Maps
API Key -> 1º Passo

Efetuar login com credenciais Google (ex. gmail) se assim solicitado

https://console.cloud.google.com/projectselector2/google/maps-apis

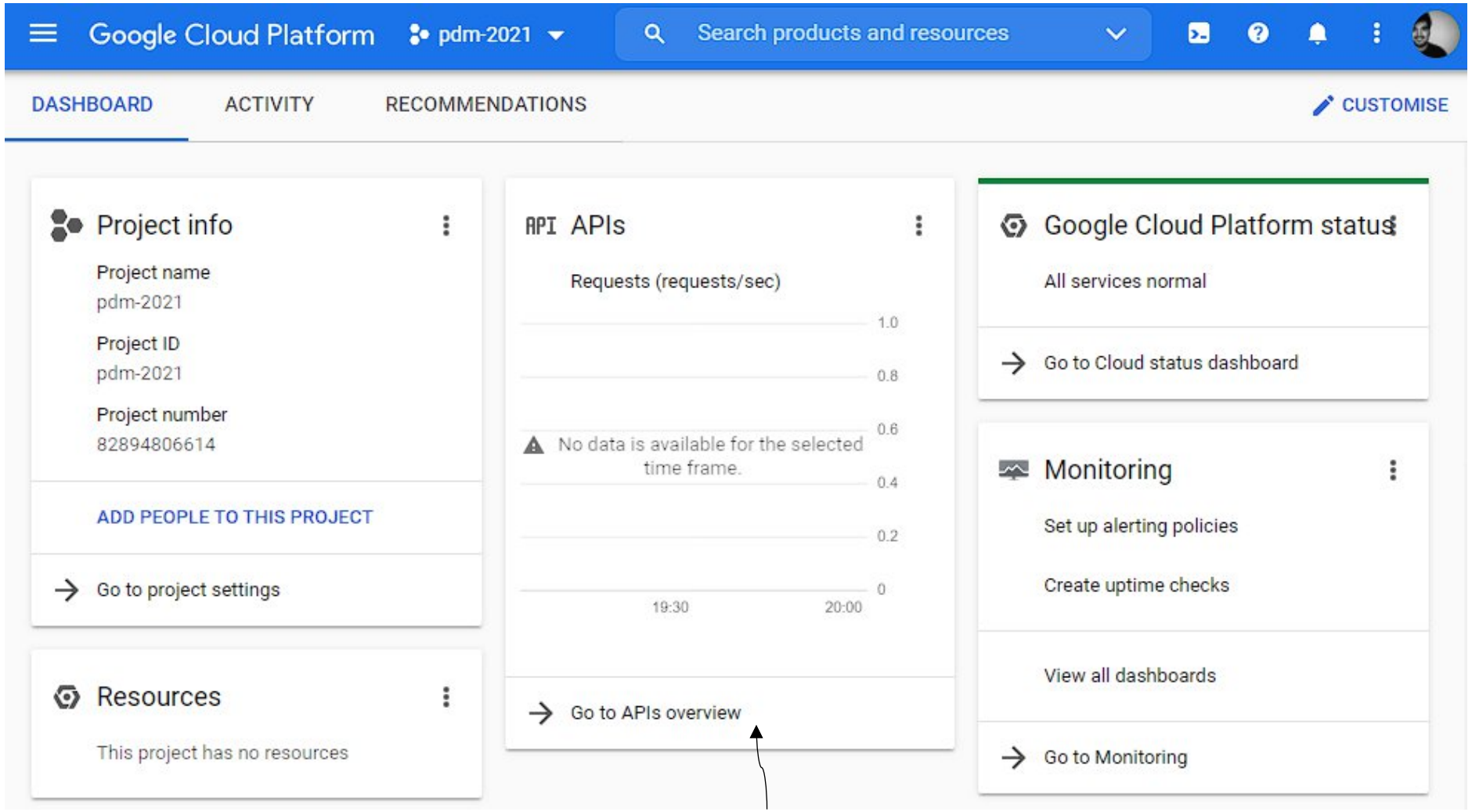


Google Maps
API Key -> 2º Passo



Google Maps

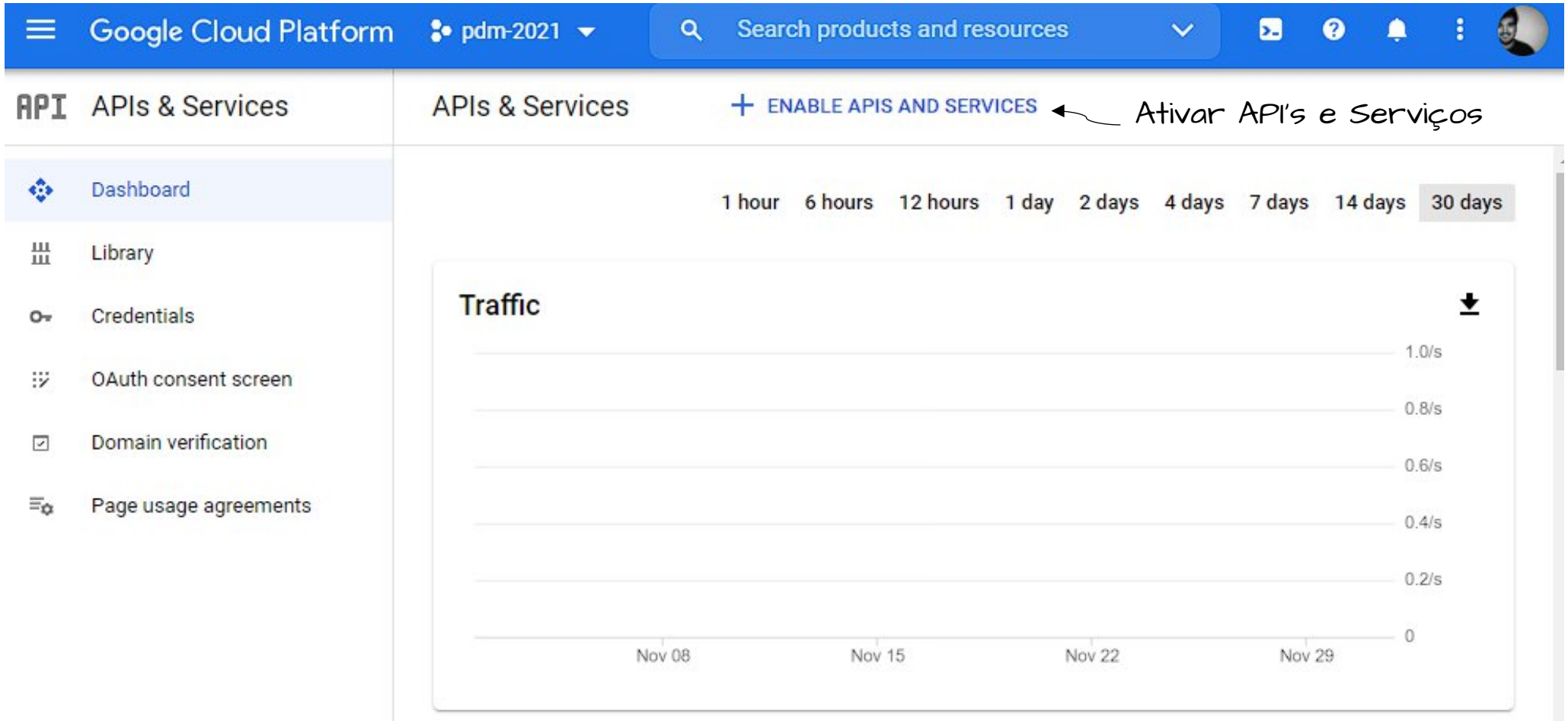
API Key -> 3º Passo



Aceder à zona de API's

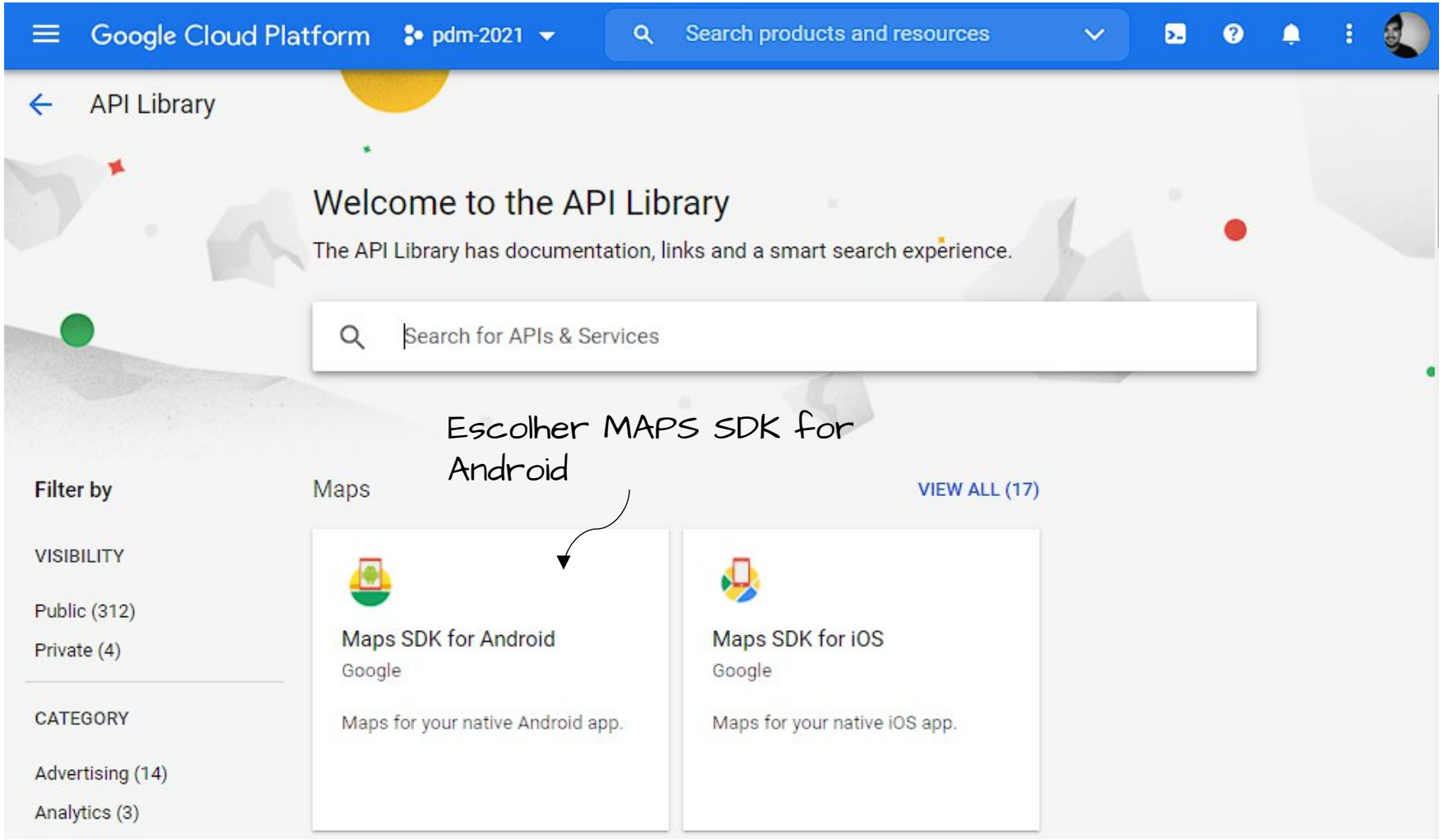
Google Maps

API Key -> 4º Passo

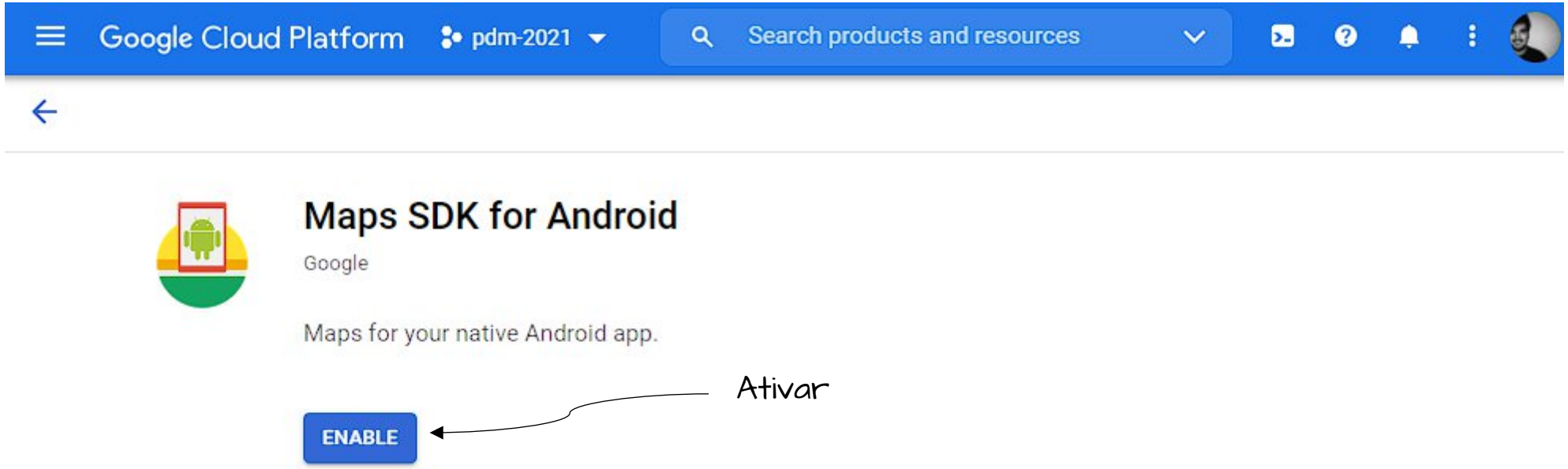


Google Maps

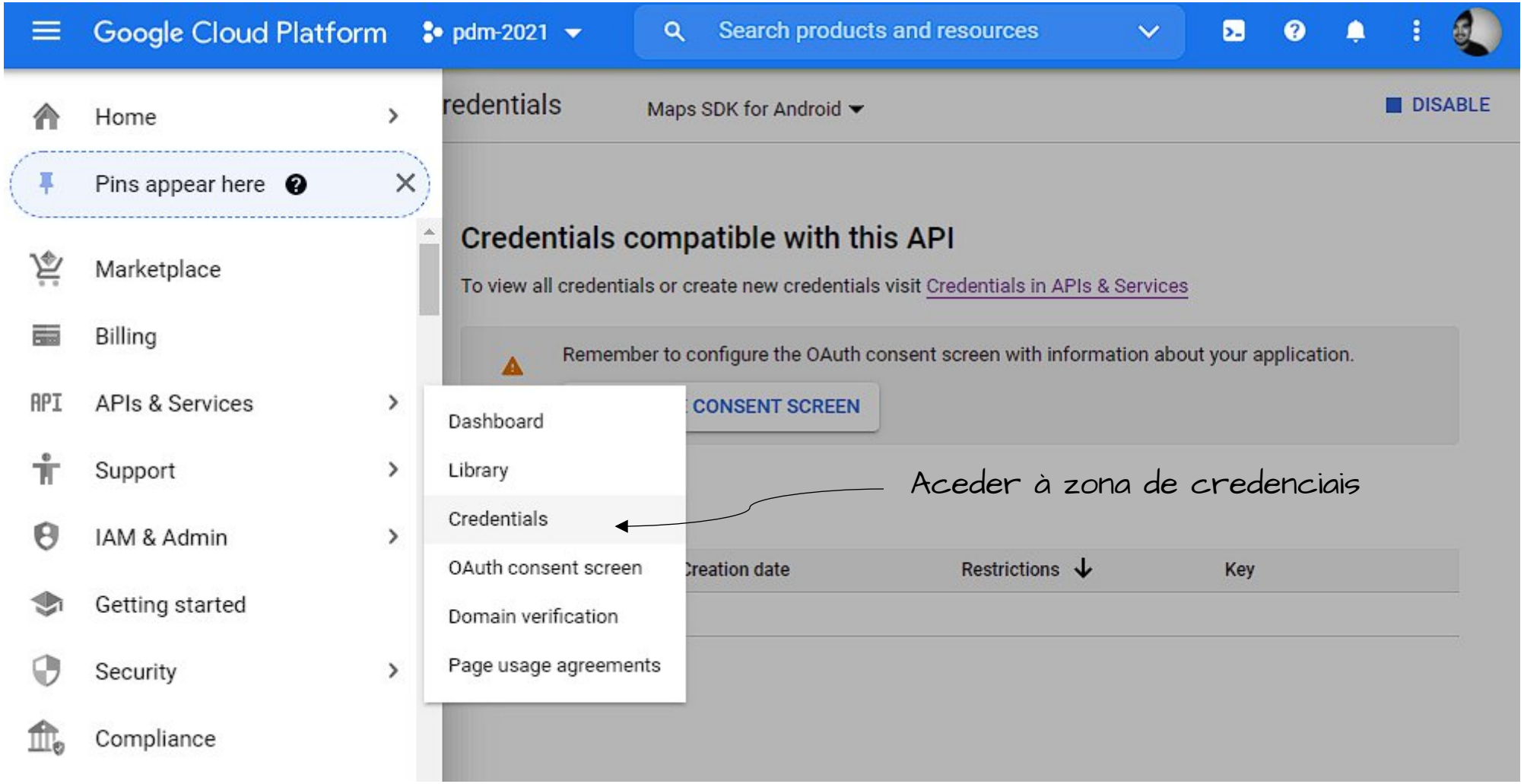
API Key -> 5º Passo



Google Maps
API Key -> 6º Passo

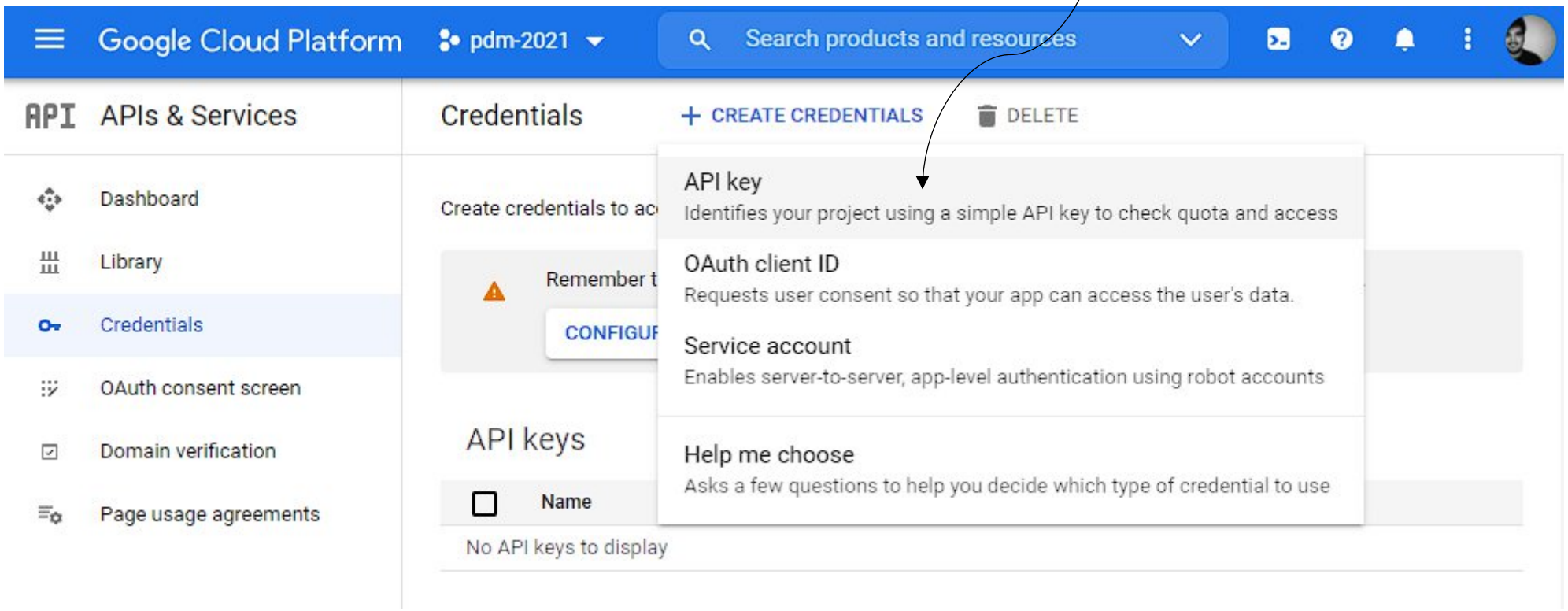


Google Maps
API Key -> 7º Passo

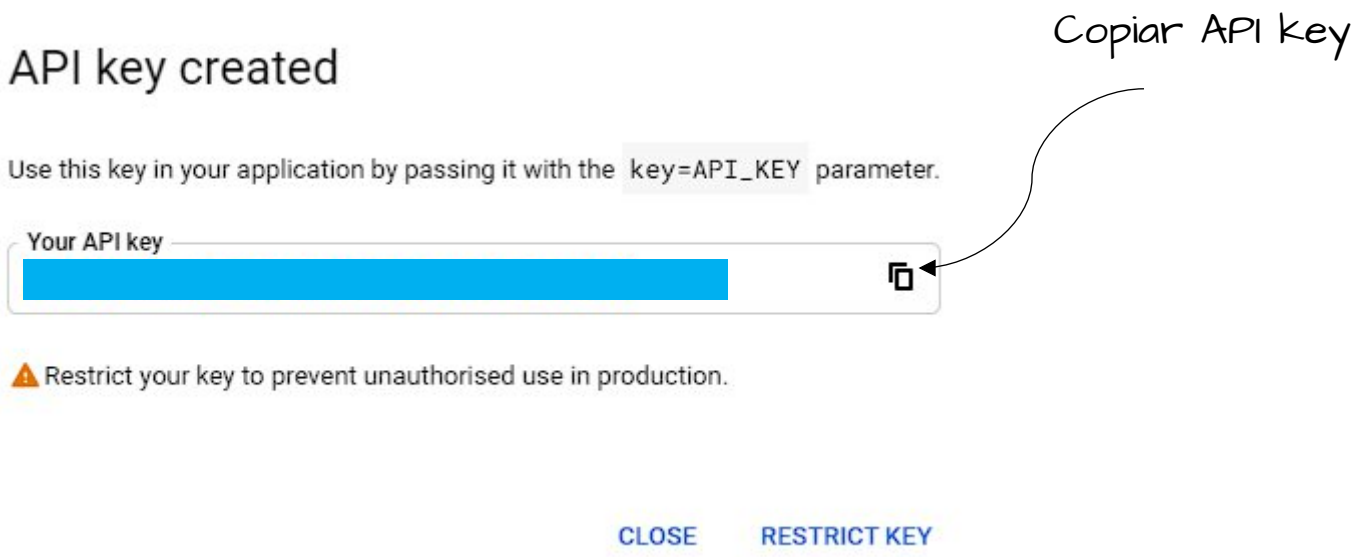


Google Maps
API Key -> 8º Passo

Adicionar credenciais



Google Maps
API Key -> 9º Passo



Google Maps

Dependências [build.gradle (Module)]

```
dependencies {

    implementation 'com.google.android.gms:play-services-maps:19.2.0'
    implementation 'com.google.android.gms:play-services-location:21.3.0'

}
```

Dependência para Google Maps

Dependência para Serviços de Localização

NORTE 2030 Programa Regional do Norte
PORTUGAL 2030
Cofinanciado pela União Europeia

Google Maps

API KEY [AndroidManifest.xml]

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="pt.ipp.estg.t_gmaps">

    <application

        <meta-data android:name="com.google.android.geo.API_KEY"
                   android:value="AAAAaaaaAAAAaaaaAAAAAaaaaaAAA" />
    </application>

</manifest>
```

Google Maps

Fragment [main_activity.xml]

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical">

    <fragment
        android:id="@+id/maps_google"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:name="com.google.android.gms.maps.SupportMapFragment“
        xmlns:map="http://schemas.android.com/apk/res-auto"
        map:cameraZoom="16"
        map:mapType="normal"
        map:uiCompass="true"
        map:uiRotateGestures="true"
        map:uiScrollGestures="true"
        map:uiTiltGestures="true"
        map:uiZoomControls="true"
        map:uiZoomGestures="true"
        />

</LinearLayout>
```

Configurações do mapa. É possivel controlar níveis de zoom, tipo de mapa, localização inicial, etc.

Google Maps

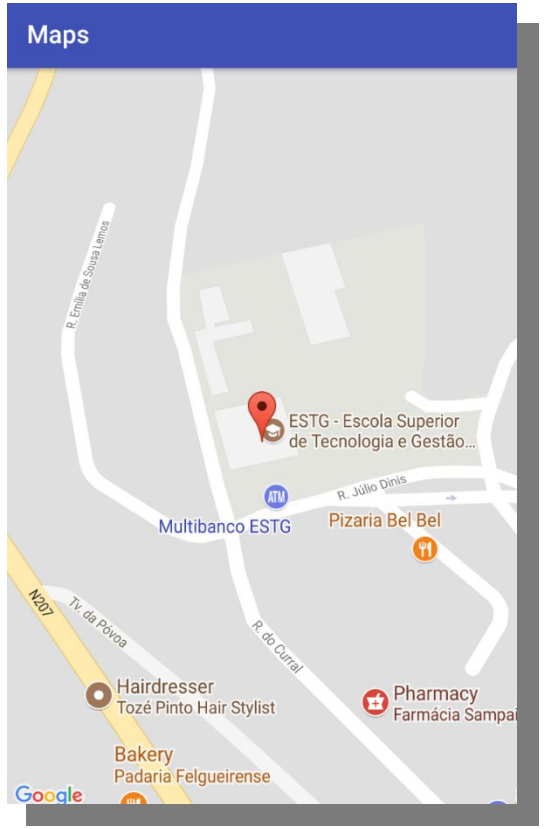
Adicionar mapa [ActivityMain.java]

Implementa OnMapReadyCallback

```
public class MainActivity extends AppCompatActivity implements OnMapReadyCallback {

    private SupportMapFragment mapFragment;
    private GoogleMap mGoogleMap;


    @Override
    protected void onCreate(Bundle savedInstanceState) {

        (…)

        mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.maps_google);
        mapFragment.getMapAsync(this);

    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mGoogleMap = googleMap;
    }
}
```

Google Maps

Adicionar Marker [ActivityMain.java]




```
private void addMarker(Location location, String title, String snippet){
    LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());

    Marker marker = mGoogleMap.addMarker(new MarkerOptions()
                            .position(latLng)
                            .title(title)
                            .snippet(snippet));
}
```

Google Maps

Marker Click [ActivityMain.java]

```
mGoogleMap.setOnMarkerClickListener(new GoogleMap.OnMarkerClickListener() {
    @Override
    public boolean onMarkerClick(Marker marker) {
        //TODO: Adicionar ação
        return false;
    }
});
```

Google Maps

Zoom para localização [ActivityMain.java]

```
private void zoomToLocation(Location location, String title, String snippet){

    LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());

    CameraUpdate cameraUpdate = CameraUpdateFactory.newLatLng(latLng);
    mGoogleMap.animateCamera(cameraUpdate);
}
```

Google Maps

Procurar localização pelo nome [ActivityMain.java]

Objeto Geocoder permite obter dados geográficos (ex. Latitude e Longitude) através de um nome de localização

```
private LatLng getLocation(String location) {

    Geocoder geocoder = new Geocoder(this);
    List<Address> addressList = new ArrayList<>();
    Address address;
    LatLng latlng = null;

    try {
        addressList = geocoder.getFromLocationName(location, 1);
    } catch (IOException ioException) {
        ioException.printStackTrace();
    }finally {
        if(addressList.size() > 0){
            address = addressList.get(0);
            latlng = new LatLng(address.getLatitude(), address.getLongitude());
            return latlng;
        }
    }

    return new LatLng(0,0);
}
```

Leitura Adicional

Setup Play Services:

https://developers.google.com/android/guides/setup

Location:

https://developer.android.com/develop/sensors-and-location/location

Google Maps for Android:

https://developers.google.com/maps/documentation/android-sdk/overview

Here Maps for Android:

https://developer.here.com/products/here-sdk

P.PORTO

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Programação Para Dispositivos Móveis I

GOOGLE PLAY SERVICES
LOCALIZAÇÃO, GOOGLE MAPS

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis
Ricardo Barbosa , rmb@estg.ipp.pt
Carlos Aldeias, cfpa@estg.ipp.pt

Adaptação do conteúdo dos slides de João Ramos jrmr@estg.ipp.pt e Fábio Silva fas@estg.ipp.pt

NORTE 2030 Programa Regional do Norte
PORTUGAL 2030
Cofinanciado pela União Europeia