



Aula 13

GPS e Google Maps

Prof. Gabriele Dani

Roteiro

- Location-based Services
- Geolocalização
- Localização no Android
 - LocationManager
 - Location
- Exemplos :
 - Localização Atual - Coordenadas
 - Localização Atual – Google Maps

Location-Based Services

- Location-Based Services ou LBS permitem ao software obter a sua localização corrente.
- Isto inclui localização obtida pelo GPS (Global Positioning System), e também pela rede de telefonia e WI-FI.

Geolocalização

- Geolocalização : Identificação da posição real no globo de um usuário final.
 - Suportado por várias aplicações Android
 - Uma das razões da popularidade dos *smartphones* atuais.
 - Possível através da combinação de receptores de ondas de rádio e algoritmos de localização programados.

Exemplos de aplicações

- São exemplos de aplicações que podem utilizar serviços de localização:
 - Compartilhamento de caronas
 - Encontrar pessoas
 - Redes sociais
 - Propaganda direcionada pela posição
 - Guias de turismo
 - Navegação
 - Gerenciamento de tráfego
 - Jogos móveis
 - Serviços de emergência

Geolocalização nos Smartphones

- Nos dispositivos móveis, a geolocalização pode ser realizada através de:
 - GPS
 - WI-FI
 - Rede Celular (EDGE, 3G, HSPDA, 4G,...)

Básico de Geolocalização

- **GPS significa Global Positioning System**

- Frota de satélites orbitando a terra a uma altura de 20000Km
- Frota composta de 24 à 32 satélites operacionais
- Períodos de órbitas de 12 horas, velocidade de 3.9 Km/s (14040Km/h).

- **Sistemas de Navegação Disponíveis:**

- **Navstar** : Operado pelo departamento de defesa americano para fins civis e militares
- **Glonass** : Operado pelas forças de defesa russas
- **Galileo** : Operado pelos EUA (em desenvolvimento)



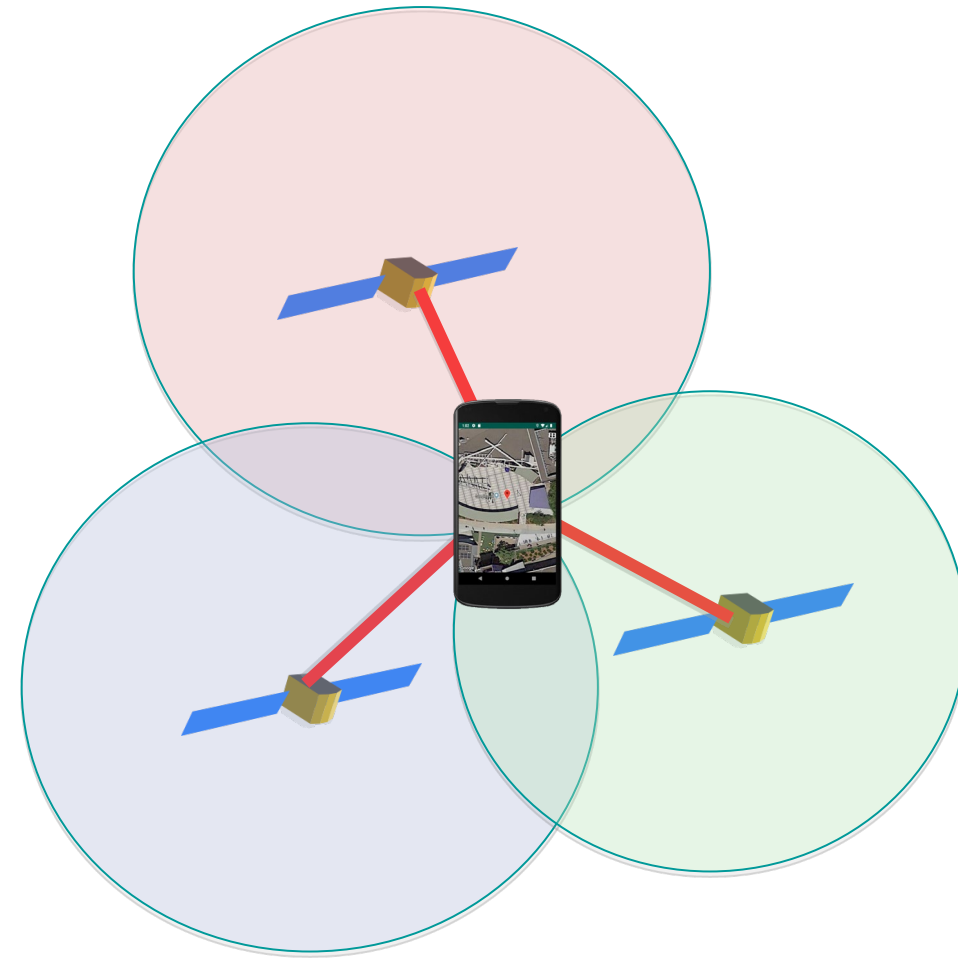
Básico de Geolocalização

- **Cada satellite transmite periodicamente:**

- Sua posição corrente
- A hora atual através de um relógio atômico

- **Operações do receptor GPS:**

- Receber os dados passivamente (não transmite nada)
- Calcular o atraso (delay) do sinal
- Através do delay calcular a distância até o satélite ($\text{distancia} = \text{delay} * c$)
- Através de múltiplas distâncias (pelo menos 3), determina a posição corrente



Satélites?

- Cada satélite transmite em duas frequências da banda **UHF**:
 - Canal L1 : dados civis
 - Sinais codificados usando Code Division Multiple Access (**CDMA**)
 - Junto com dados/localização, cada satélite transmite dados chamados almanaque, que são dados das rotas orbitais dos satélites.
 - Através do almanaque, o receptor de GPS sabe quais são os satélites visíveis em sua posição.
- **Problema** : Quando o GPS inicia, há uma demora para localizar as posições iniciais dos satélites e receber o almanaque completo
- **Solução** : Uso de tecnologia Assisted-GPS (**A-GPS**)

Assisted GPS (A-GPS)

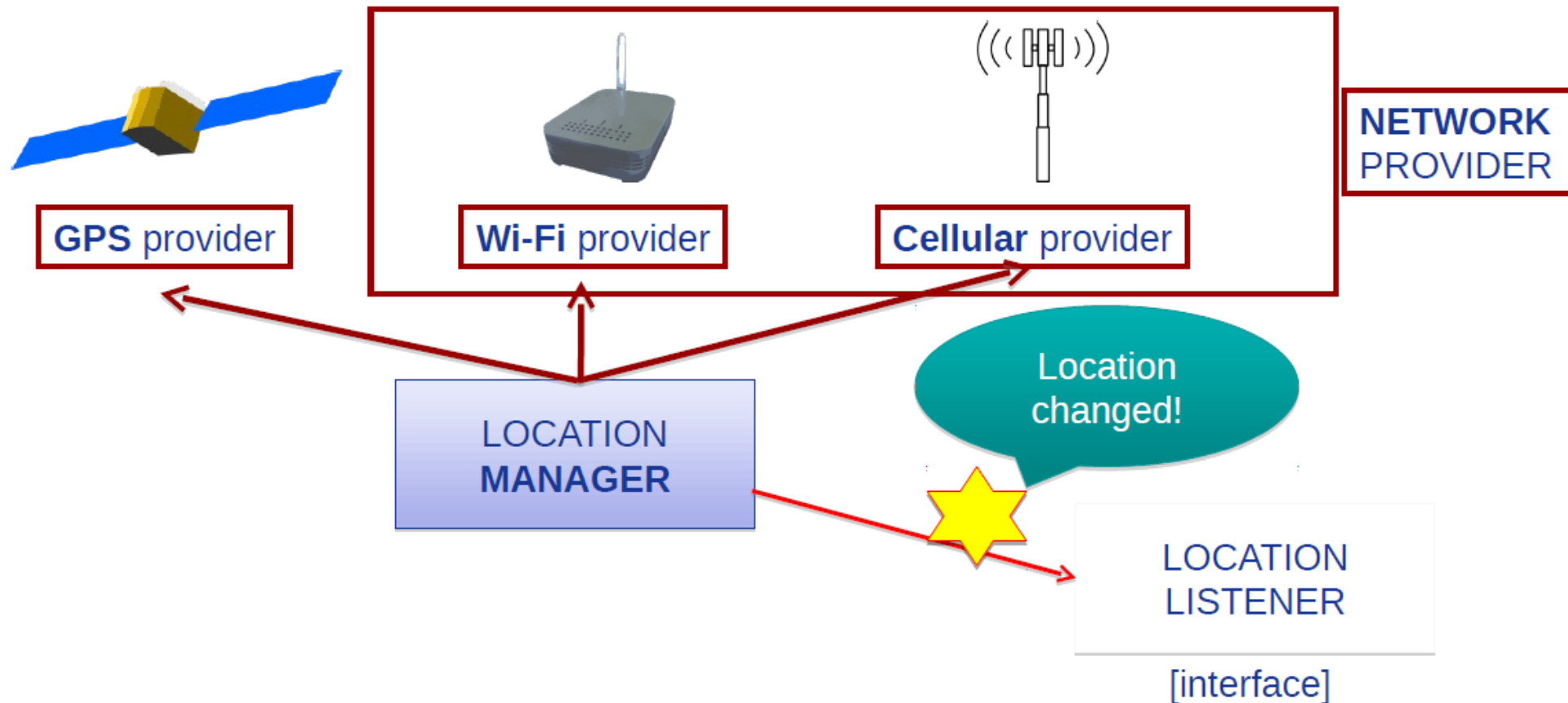
- Dispositivos móveis usam, além do GPS, outras técnicas para melhorar a precisão da posição calculada:
 - Descubrem a qual antena de celular o aparelho está vinculado
 - Descubrem em qual rede WI-FI o celular está
 - Nesse caso, usa-se a localização WI-FI

Localização WI-FI

- A localização por WI-FI é realizada através de triangulação ou através da identificação da estação de rádio (técnica usada pelo Android)
 - O **Smartphone** liga a interface WI-FI, e detecta o endereço **MAC** e a **SSID** dos roteadores WI-FI em seu alcance
 - Com esses dados, faz uma consulta ao **serviço de localização do Google**
 - Este, baseado em informações gravadas sobre redes WI-FI conhecidas, fornece ajuda para determinar a posição corrente do aparelho
- Quem popula o banco de dados do Google? Os usuários, quando habilitam o serviço de localização.

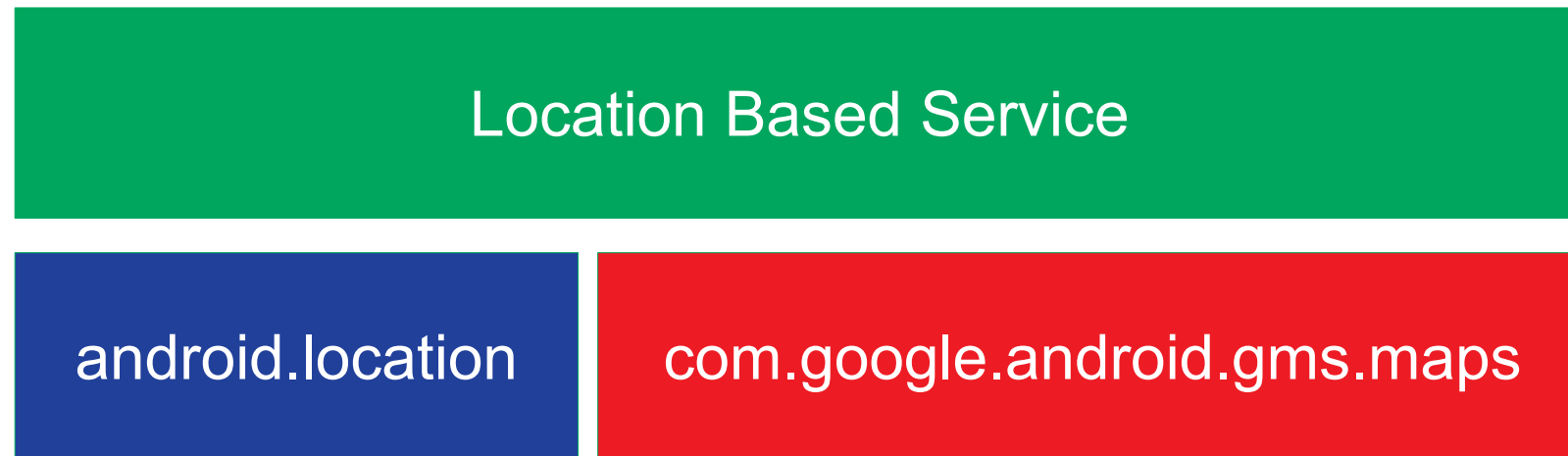
Localização WI-FI

- Como obter a localização do dispositivo:



LBS API

- A API LBS contém dois pacotes ***android.location*** e ***com.google.android.gms.maps*** que fornecem uma visão inicial do suporte contido na plataforma Android para a construção de serviços baseados em localização



LocationManager

- A classe ***LocationManager*** fornece uma API para determinar a localização e orientação, se o dispositivo suportar.
- Essa classe dá acesso aos serviços do sistema de localização que permitem às aplicações obter atualizações periódicas da posição geográfica do aparelho, ou disparar uma aplicação específica quando o dispositivo entrar nas redondezas de uma dada posição.
- A classe ***LocationManager*** não deve ser instanciada diretamente, devendo o programador solicitar uma referência a ela para manipulá-la.

LocationManager

- Uma aplicação com ***LocationManager*** pode fazer três coisas:
 - Consultar a lista de ***LocationProviders*** conhecidos pelo ***LocationManager*** para obter a última posição conhecida do dispositivo
 - Registrar ***listeners*** para observar alterações periódicas de posição
 - Registrar um determinado ***Intent*** para ser disparado se o dispositivo aproximar-se de uma determinada posição (Latitude e Longitude), com o raio de distância calculado em metros

Location

- A classe ***Location*** representa uma posição geográfica percebida em um dado momento. Um objeto ***location*** possui uma latitude, uma longitude, um momento UTC (quando a posição foi obtida), e opcionalmente a altitude e velocidade.

Exemplos de uso

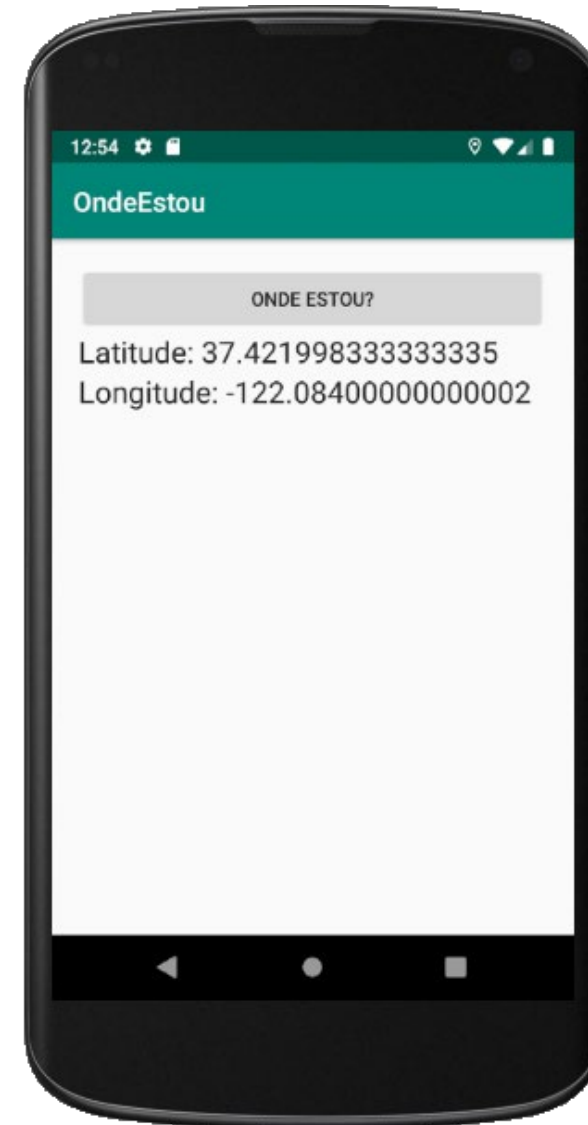
- Localização atual por Coordenadas (**OndeEstou**)
- Localização atual mostrada no Google Maps (**MapasSimples**)

Projeto OndeEstou

- O projeto OndeEstou mostra um exemplo de obtenção de Coordenadas usando o LocationManager, sem exibir mapas
- Esse tipo de aplicação pode ser útil para registrar pontos visitados, por exemplo

Projeto OndeEstou

- Criaremos um pequeno aplicativo como mostrado ao lado.
- Após o botão “Onde Estou?” ser pressionado, as coordenadas são exibidas e atualizadas à cada alteração de posição



AndroidManifest.xml

- Para que o aplicativo possa utilizar o GPS, é necessário editar o arquivo ***AndroidManifest.xml*** e adicionar as seguintes linhas:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />  
<uses-permission-sdk-23 android:name="android.permission.ACCESS_FINE_LOCATION" />
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    private Button btnGps;  
    private TextView txtLatitude, txtLongitude;  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        txtLatitude = (TextView) findViewById(R.id.txtLatitude);  
        txtLongitude = (TextView) findViewById(R.id.txtLongitude);  
  
        btnGps = (Button) findViewById(R.id.btnGps);  
        btnGps.setOnClickListener(new Button.OnClickListener() {  
            public void onClick(View v) {  
                pedirPermissoes();  
            }  
        });  
    }  
  
    ...  
}
```

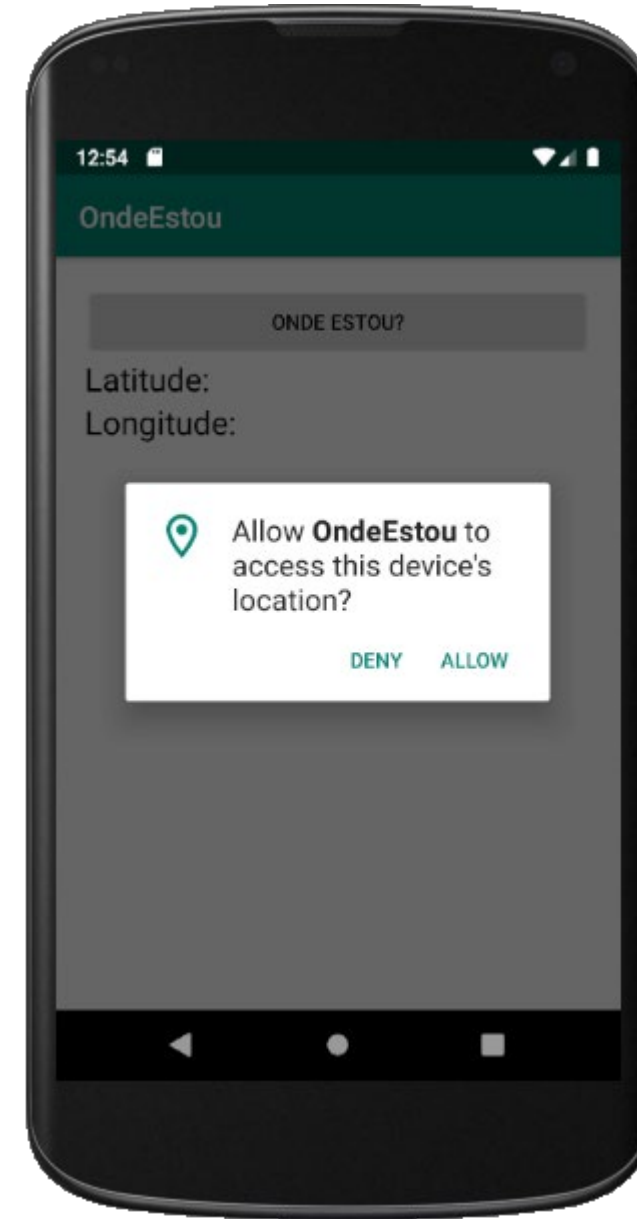
pedirPermissoes(...)

```
private void pedirPermissoes() {  
  
    if (ActivityCompat.checkSelfPermission(  
        this, Manifest.permission.ACCESS_FINE_LOCATION) !=  
        PackageManager.PERMISSION_GRANTED  
        && ActivityCompat.checkSelfPermission(  
        this, Manifest.permission.ACCESS_COARSE_LOCATION)  
        != PackageManager.PERMISSION_GRANTED) {  
  
        ActivityCompat.requestPermissions(this, new String[]  
            {Manifest.permission.ACCESS_FINE_LOCATION}, 1);  
    }  
    else  
        configurarServico();  
}
```

- É necessário pedir autorização para usar o GPS. Caso o aplicativo não tenha a permissão necessária (o que somente pode acontecer na versão 6+ do Android), chamamos um método padrão do Android para requisitar a permissão que queremos, o ***ActivityCompat.requestPermissions***

Permissões

- O aplicativo solicitará permissão de uso do GPS em seu primeiro uso



onRequestPermissionsResult(...)

```
@Override
public void onRequestPermissionsResult(int requestCode,
                                     String permissions[], int[] grantResults) {
    switch (requestCode) {
        case 1: {
            if (grantResults.length > 0 && grantResults[0] ==
                PackageManager.PERMISSION_GRANTED) {
                configurarServico();
            } else {
                Toast.makeText(this, "Não vai funcionar!!!", Toast.LENGTH_LONG).show();
            }
            return;
        }
    }
}
```

- Do Android 6 em diante é possível alterar as permissões após concedê-las. Só que se a aplicação não puder usar o GPS, nada funcionará. Nesse caso, mostramos um ***Toast*** para o usuário.

configurarServico(...)

```
public void configurarServico(){
    try {
        LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

        LocationListener locationListener = new LocationListener() {
            public void onLocationChanged(Location location) {
                atualizar(location);
            }

            public void onStatusChanged(String provider, int status, Bundle extras) { }

            public void onProviderEnabled(String provider) { }

            public void onProviderDisabled(String provider) { }
        };
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, locationListener);
    } catch (SecurityException ex){
        Toast.makeText(this, ex.getMessage(), Toast.LENGTH_LONG).show();
    }
}
```



Criação do **Listener**

- Nesse método instanciamos o ***LocationManager*** e registramos o ***Listener*** que irá atualizar a tela a cada mudança de localização do aparelho.

atualizar(...)

```
public void atualizar(Location location)
{
    Double latPoint = location.getLatitude();
    Double lngPoint = location.getLongitude();

    txtLatitude.setText(latPoint.toString());
    txtLongitude.setText(lngPoint.toString());
}
}
```

- Por último, define-se o método ***atualizar(...)***, que apresenta nos campos da tela as novas coordenadas geográficas obtidas após a atualização da posição do aparelho.

Layout da aplicação

- O aplicativo **OndeEstou** utiliza um layout simples, linear que é demonstrado nos próximos slides.
- As distâncias das margens foram externalizadas para um arquivo de dimensões (**dimen.xml**), também demonstrado.

activity_main.xml (1/2)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btnGps"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Onde estou?"/>

    <LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
```

activity_main.xml (2/2)

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="@android:style/TextAppearance.DeviceDefault.Large"
    android:text="Latitude: "/>

<TextView
    android:id="@+id/txtLatitude"
    android:textAppearance="@android:style/TextAppearance.DeviceDefault.Large"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
</LinearLayout>

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="@android:style/TextAppearance.DeviceDefault.Large"
        android:text="Longitude: "/>

    <TextView
        android:id="@+id/txtLongitude"
        android:textAppearance="@android:style/TextAppearance.DeviceDefault.Large"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
</LinearLayout>
```

values/dimen.xml

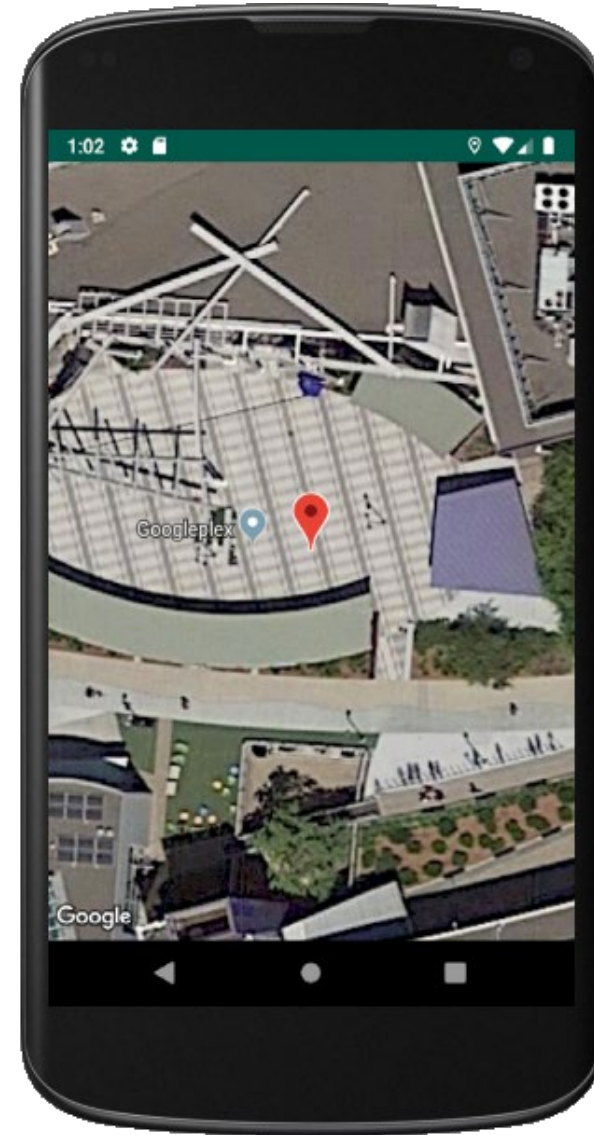
```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <dimen name="activity_vertical_margin">20dp</dimen>  
  <dimen name="activity_horizontal_margin">20dp</dimen>  
</resources>
```

API do Google Maps

- O próximo exemplo usa a API do ***Google Maps*** para exibir o mapa com a posição atual do dispositivo
- Para usar essa API, é necessário solicitar uma chave de API no site da Google

Projeto MapasSimples

- Esse exemplo mostra a posição atual do dispositivo em uma visão de satélite, numa ***View Google Maps***



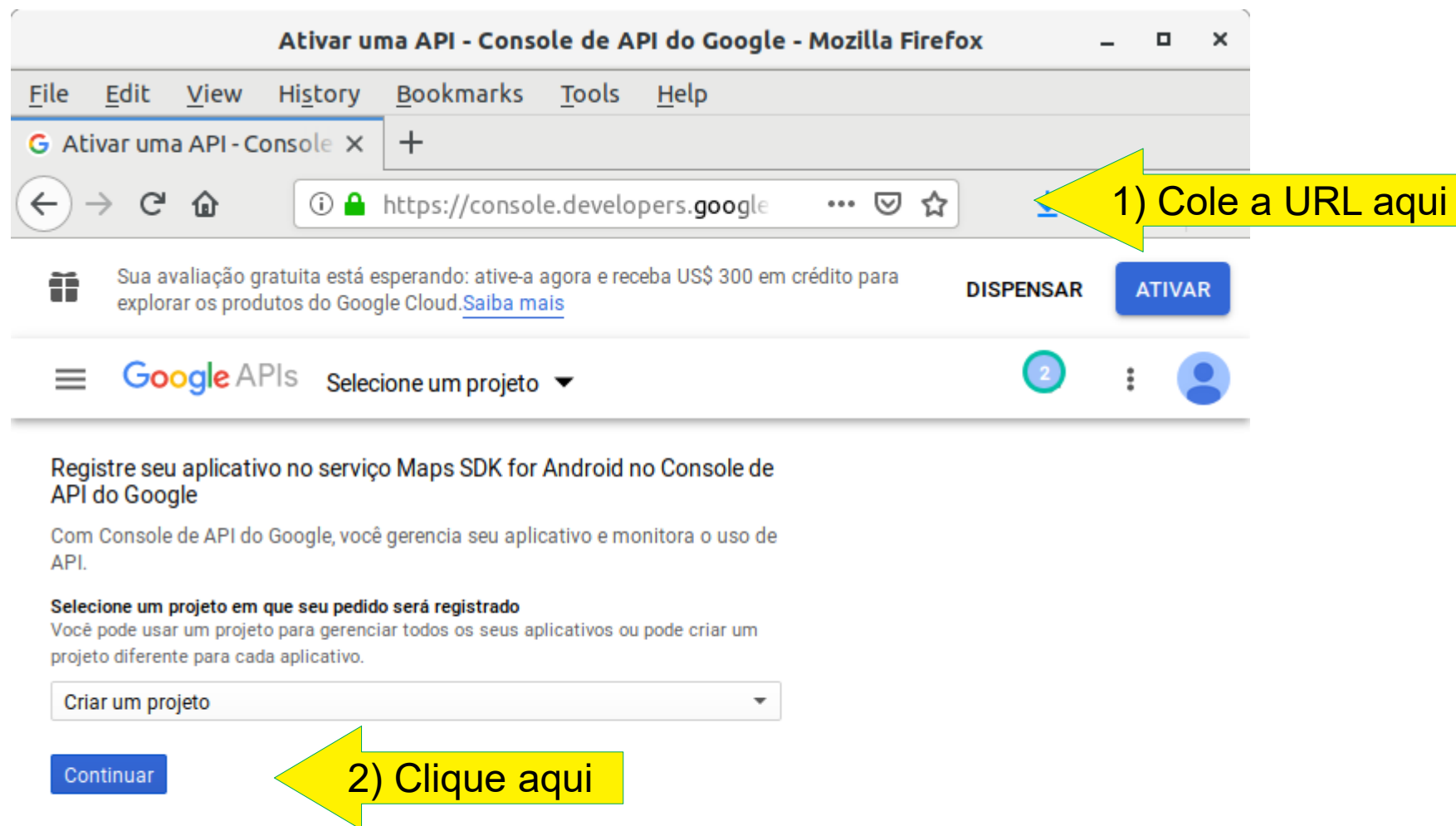
Criação do Projeto

- Crie um projeto chamado ***MapasSimples***, com uma ***Activity*** do tipo ***Google Maps Activity***
- Nesse caso, o projeto criado usará não uma ***view***, mas um ***SupportMapFragment***, que é um componente Android mais abrangente do que uma ***view***.

Chave da API

- Encontre o arquivo *values/google_maps_api.xml*
- Dentro dele, você encontrará a url para obter a chave. Siga os passos indicados na página, obtenha a chave e a substitua onde estiver indicado com ***YOUR_KEY_HERE***

Criação da chave da API



The screenshot shows the 'Ativar uma API - Console de API do Google' page in a Mozilla Firefox browser. The address bar shows the URL `https://console.developers.google`. A yellow arrow points to the address bar with the text '1) Cole a URL aqui'. Below the browser window, the page content includes a message about a free evaluation, a 'Selecione um projeto' dropdown menu, and a 'Continuar' button. A second yellow arrow points to the 'Continuar' button with the text '2) Clique aqui'.

Ativar uma API - Console de API do Google - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Ativar uma API - Console X +

https://console.developers.google

Sua avaliação gratuita está esperando: ative-a agora e receba US\$ 300 em crédito para explorar os produtos do Google Cloud. [Saiba mais](#) DISPENSAR ATIVAR

Google APIs Selecione um projeto 2

Registre seu aplicativo no serviço Maps SDK for Android no Console de API do Google

Com Console de API do Google, você gerencia seu aplicativo e monitora o uso de API.

Selecione um projeto em que seu pedido será registrado

Você pode usar um projeto para gerenciar todos os seus aplicativos ou pode criar um projeto diferente para cada aplicativo.

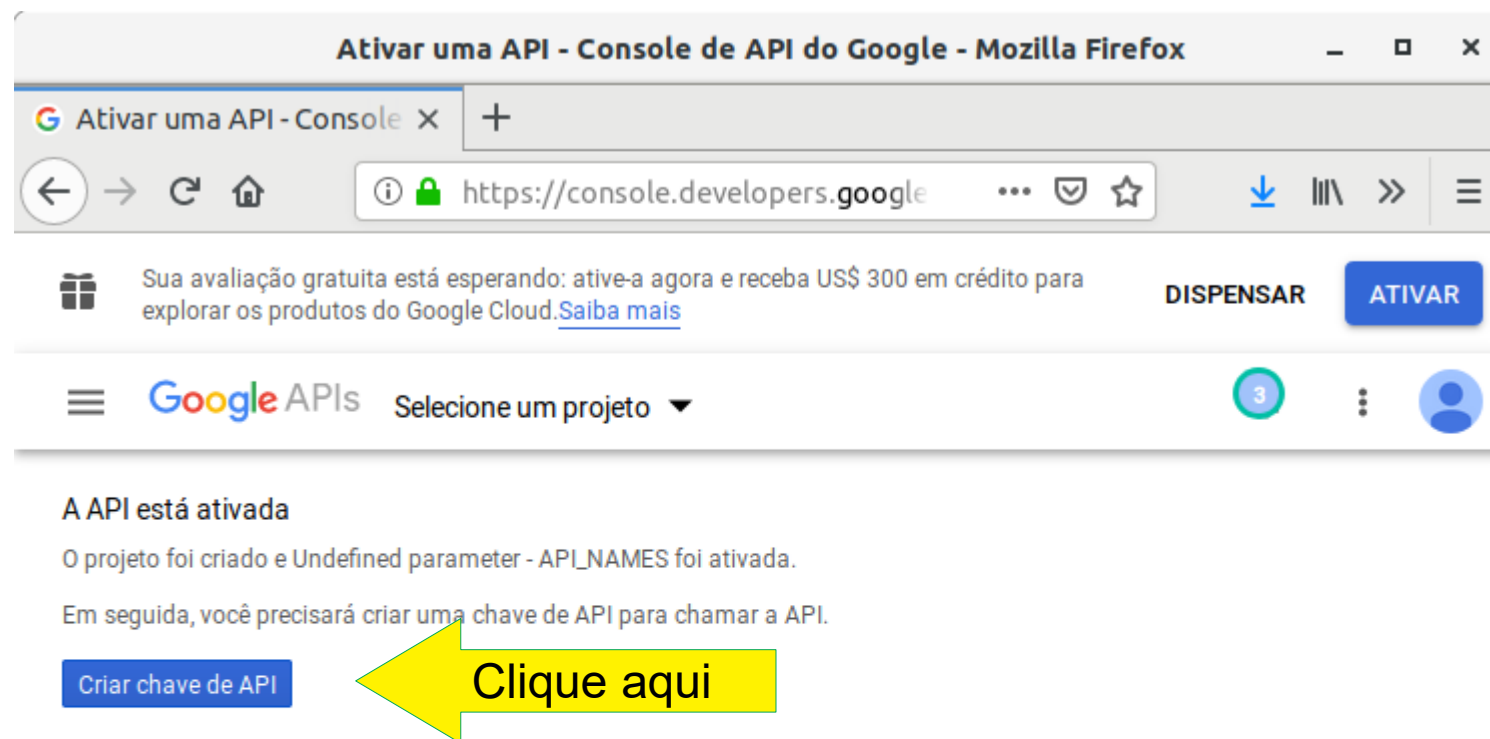
Criar um projeto

Continuar

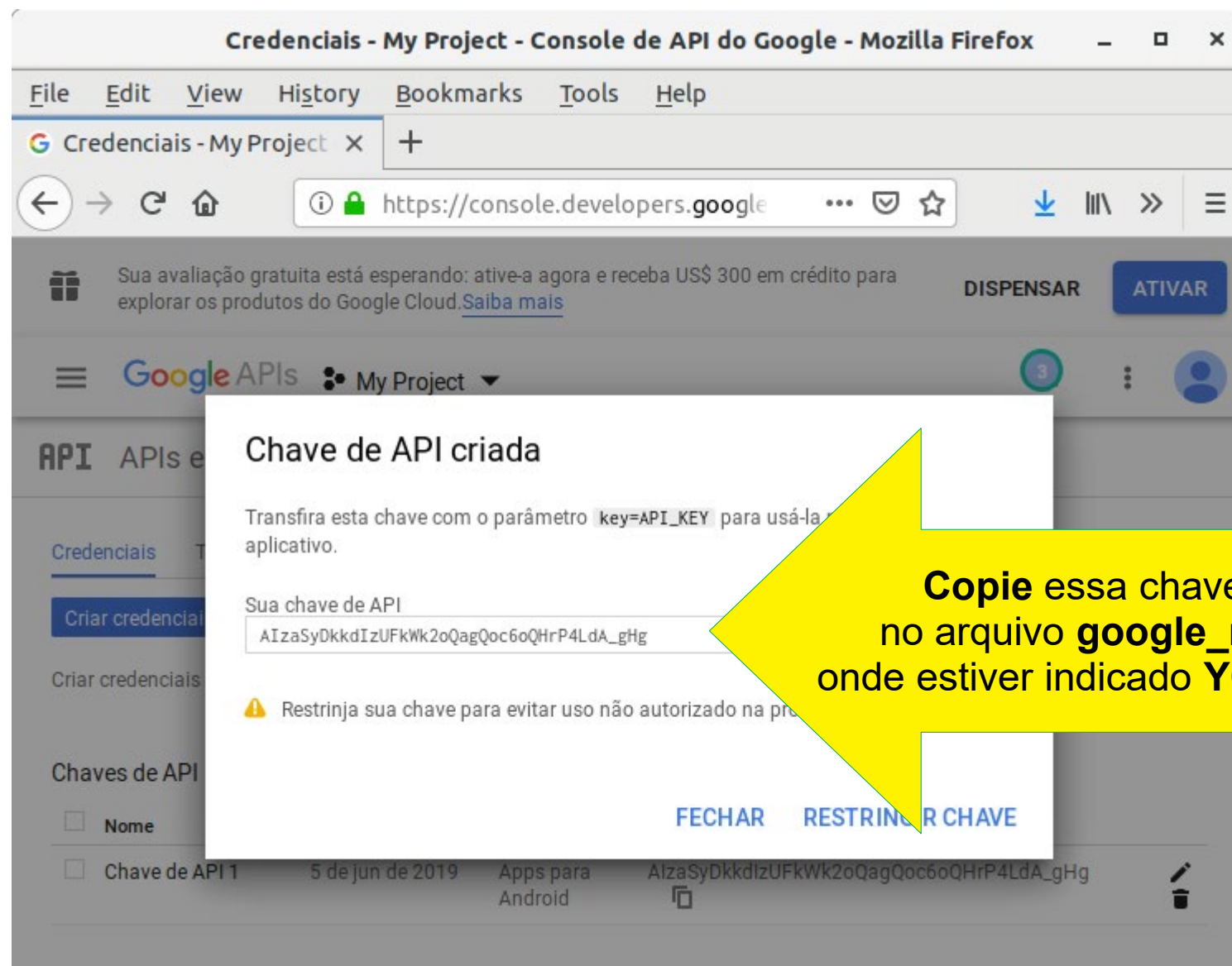
1) Cole a URL aqui

2) Clique aqui

Criação da chave da API



Criação da chave da API



**Copie essa chave e substitua
no arquivo `google_maps_api.xml`
onde estiver indicado **YOUR_KEY_HERE****

AndroidManifest.xml

- Novamente, para que o aplicativo possa utilizar o GPS, é necessário editar o arquivo ***AndroidManifest.xml*** e adicionar as seguintes linhas:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />  
<uses-permission-sdk-23 android:name="android.permission.ACCESS_FINE_LOCATION" />
```

MapsActivity.java

- Note que esse projeto usa uma **activity** com nome diferente. Porém a implementação inicial é semelhante:

```
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {  
  
    private GoogleMap mMap;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_maps);  
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.  
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()  
            .findFragmentById(R.id.map);  
        mapFragment.getMapAsync(this);  
  
        pedirPermissoes();  
    }  
}
```

MapsActivity.java

- Alguns métodos são iguais ao projeto **OndeEstou** e podem ser copiados de lá:
 - pedirPermissoes()
 - onPermissionsResult(...)
 - configurarServico()
- Outros devem ser implementados:

Método atualizar(...)

- O método atualizar marca sua posição no mapa, posicionando a "câmera" sobre a posição:

```
public void atualizar(Location location)
{
    LatLng posicaoLocal = new LatLng(location.getLatitude(),location.getLongitude());
    mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(posicaoLocal, 20.0f));
    mMap.addMarker(new MarkerOptions().position(posicaoLocal).title("Você está aqui"));
}
```

Método onMapReady(...)

- Por fim o método onMapReady(...) faz o posicionamento inicial do mapa sobre a cidade de Caxias do Sul, assim que o mapa é aberto:

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);

    LatLng caxias = new LatLng(-29.167, -51.179);
    mMap.addMarker(new MarkerOptions().position(caxias).title("Caxias do Sul"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(caxias));
}
```

Referências

- Reto Meier, Ian Lake - **Professional Android** (2018, Wrox)
- <https://www.luiztools.com.br/post/como-criar-um-aplicativo-android-com-gps/>
- <https://blog.caelum.com.br/usando-o-google-maps-e-gps-no-android/>
- Luca Bedogni, Marco Di Felice - Programming with Android – Google Maps Library