Construindo MapaActiviry

Já conseguimos acessar o mapa sem acessar o Google Maps externamente. Agora, queremos customizá-lo da melhor forma possível.

Pedimos para o Android Studio criar a Maps, no entanto, não vemos o que está acontecendo dentro para que isto aconteça. Para entendermos a forma de colocarmos o mapa na tela, nós iremos destruir o MapsActivity.java e vamos construir o processo mostrando o passo a passo.

Primeiramente, dentro do pacote padrão iremos apagar o arquivo já existente. Certifique-se que o layout, o arquivo activity\_maps.xml foi apagado também. Abriremos uma nova Activity, de acordo com o processo que foi mostrado anteriormente. Iremos chama-la de MapaActivity.java. Por padrão, ela irá trazer o método onCreate() e utilizará o layout activity\_mapa.

O segredo de usarmos uma mapa dentro da aplicação é que estamos reaproveitando um comportamento de parte de outra aplicação, o que isto nos lembra? De **fragments**. Estamos aproveitando um Fragment que já existe no Android e que conseguimos colocar dentro das nossas Activities.

Para recapitularmos, se quiséssemos criar uma Activity em que a tela inteira fosse ocupada por um Fragment, qual procedimento teremos que seguir? Vamos abrir o activity\_mapa.xml. Apagaremos o RelativeLayoutcriado automaticamente e reservaremos um espaço que será substituído pelo Fragment. Neste caso, usaremos o FrameLayout e especificaremos que ele deve ocupar a tela inteira, com o match\_parent.

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/frame\_mapa"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 />

O FrameLayout é uma moldura que será substituída pelo Fragment. Por isso, adicionamos um id(frame\_mapa), que será usado mais adiante.

Voltaremos para o MapaActivity.java, que atualmente está assim:

package br.com.alura.agenda;  
  
import ...  
  
public class MapaActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_mapa);  
 }  
}

Como faremos para incluir o Fragment? Usando o método getSupportFragmentManager() que irá devolver uma referência que guardaremos em uma variável que chamaremos de manager.

FragmentManager manager = getSupportFragmentManager();

Qualquer operação que executamos com o Fragment no Android precisa de uma transação. Guardaremos o beginTransaction dentro da variável local tx. Agora, é possível realizar a operação. Logo, adicionaremos um tx.replace e especificaremos o frame que queremos subtituir e qual será o fragment que irá substitui-lo.

FragmentManager manager = getSupportFragmentManager();  
FragmentTransaction tx = manager.beginTransaction();  
tx.replace(R.id.frame\_mapa, new SupportMapFragment());

Nós usamos o SupportMapFragment, que possui todas as funcionalidade de mapa do Google Maps. Para efetuar a transação, basta efetuar o tx.commit().

FragmentManager manager = getSupportFragmentManager();  
FragmentTransaction tx = manager.beginTransaction();  
tx.replace(R.id.frame\_mapa, new SupportMapFragment());  
tx.commit();

O próximo passo é mudar o menu para direcioná-lo para a nova Activity. Iremos alterar isto no ListaAlunosActivity.java, no trecho em que chamávamos o MapsActivity.

\...  
case R.id.menu\_mapa:  
 Intent vaiParaMapa = new Intent(this, MapsActivity.class);  
 startActivity(vaiParaMapa);  
 break;

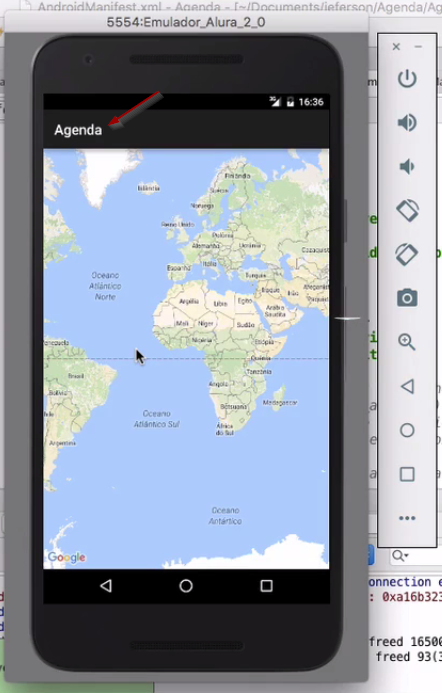
Após mudá-lo, com o MapaActivity, o código ficará assim:

@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.menu\_enviar\_notas:  
 new EnviaAlunosTask(this).execute();  
 break;  
 case R.id.menu\_baixar\_provas:  
 Intent vaiParaProvas = new Intent(this, ProvasActivity.class)  
 startActivity(vaiParaProvas);  
 break;  
 case R.id.menu\_mapa:  
 Intent vaiParaMapa = new Intent(this, MapaActivity.class);  
 startActivity(vaiParaMapa);  
 break;  
  
 }  
 return super.onOptionItemSelected(item);  
 }

Se formos conferir no AndroidManifest, já estará registrada a Activity.

\...  
<meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="AIzaSyD1X7eD4FsHo1HUsUT-Zbc1PQZVFpM0Kr4" />  
<activity android:name".MapaActivity"></activity>

Vamos rodar a aplicação e ver se está tudo funcionando.



Observe que o mapa ainda não terá um pino, nem centralizou a localização em um país específico. Isto aconteceu, porque estamos construindo a Activity do zero. Agora, também temos a barra da nossa aplicação logo acima, porque o tema da nossa aplicação está sendo usado, e ele possui uma ActionBar.

Se quiséssemos que o mapa ocupasse apenas uma parte da tela, bastava alterar as dimensões da altura e da largura no activity\_mapa.xml.

O próximo passo será customizar o mapa e adicionar os pinos nas localizações dos alunos.