

# Minicurso Spring e Android

Criado por Gabriel Schaidhauer - Aula 6



# Listas

# Listas

Listas são um recurso bastante importante ao criar uma aplicação. Sempre há dados a ser exibidos em forma de lista. No Android existe um componente específico para trabalhar com listas, que é o `ListView`.

# ListView

ListView é um componente do framework Android, que tem o objetivo de exibir listas. Este componente trabalha com base em Adapters, e a partir destes adapters popula a lista utilizando uma fonte de dados.

# Adapter

Um Adapter é uma classe responsável por alimentar uma lista. Existem alguns adapters prontos no framework Android, como é o caso do ArrayAdapter, o qual permite a exibição simples de itens através do método toString.

# Adapter

Podem ser criados Adapters personalizados herdando a classe `BaseAdapter`, que possui os métodos necessários para trabalhar com um `ListView`, e permite que possamos definir qual o layout que cada item da lista possuirá.

## ListView, Adapter?



# ListView

ListView é um componente simples, que pode ser inserido dentro de um layout.

Para definirmos o seu conteúdo como dito anteriormente usaremos um adapter. Para tanto foi adicionado um ListAdater.

Também, definimos um onItemClickListener a fim de poder capturar qual o item que foi clicado.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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="com.example.gabrielschaidhauer.aula6.ListActivity">

    <ListView
        android:id="@+id/list_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</RelativeLayout>
```

```
public class ListActivity extends AppCompatActivity {
    private ListView listView;
    private String [] nomes = {"Gabriel", "Rodrigo", "Lucas", "Nelson", "Guilherme", "Aline"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list);

        this.listView = findViewById(R.id.list_view);
        this.listView.setOnItemClickListener((parent, view, position, id) -> {
            Intent i = new Intent( packageContext: ListActivity.this, MapsActivity.class);
            i.putExtra( name: "nome", nomes[position]);
            startActivity(i);
        });
    }
    @Override
    protected void onStart() {
        super.onStart();
        this.listView.setAdapter(new ListAdapter());
    }
}
```



Tá, beleza, mas e o tal do adapter?



# ListAdapter

O ListAdapter é uma classe que criamos, com o objetivo de que cada elemento de uma lista seja exibido em um Fragment.

Este Fragment é um simples BaseAdapter, o qual possui métodos específicos para trabalhar com a lista.

O método mais importante aqui é o getView. É neste método que podemos informar a view que será exibida para cada item da lista.

```
private class ListAdapter extends BaseAdapter {

    @Override
    public int getCount() {
        return nomes.length;
    }

    @Override
    public Object getItem(int position) {
        return nomes[position];
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View olderItem, ViewGroup parent) {
        View view = null;

        if(olderItem == null) {
            view = LayoutInflater.from(ListActivity.this).inflate(R.layout.fragment_nome, parent, attachToRoot: false);
        } else {
            view = olderItem;
        }

        ((TextView) view.findViewById(R.id.nome)).setText((String) getItem(position));

        return view;
    }
}
```

# Porquê é importante saber usar listas?

Listas são um elemento corriqueiro ao longo da implementação de um sistema, entretanto, dependendo do seu tamanho, elas podem acabar representando um problema para a aplicação. Pensando nisso, o Android utiliza listas reutilizáveis, ou seja, ele só renderiza os elementos que podem ser visualizados na tela. quando um elemento deixa de estar visível, o elemento (fragment) é reaproveitado e tem seu conteúdo alterado. É por isso que não instanciamos um elemento novo em certos casos no nosso ListAdapter

# Mapas

# Mapas

Hoje, o uso de mapas está ligado a qualquer elemento que precise demonstrar um local, seja para melhor ambientar um usuário ou para permitir recursos de navegação, é inegável a importância dos mapas.

# Mapas

No Android, a api padrão para trabalhar com mapas é o Google Maps, existindo inclusive um facilitador de criação para esta activity.

Esta activity não é como uma atividade comum, ela possui alguns requisitos especiais, tais como possuir uma chave de acesso.



# Mapas

Primeiro deve-se acessar em:

File > New > Activity > Gallery > Google Maps Activity.

Depois de clicar isso, nomeie e crie a activity.

Após isto, são criados 3 novos recursos. uma Activity, uma Activity class e um arquivo xml da api, onde deve ser inserida a chave obtida para uso da api.

```
<resources>
  <!--
    TODO: Before you run your application, you need a Google Maps API key.
  -->
  <string
    name="google_maps_key"
    templateMergeStrategy="preserve"
    translatable="false">AIzaSyBUj1EIHnAwjihadUCMLrrPo_NhY2W2RhY</string>
</resources>
```

application, you need a Google Maps API key. the directions and press "Create" at the end:

[https://console.developers.google.com/flows/enableapi?apiid=maps\\_android\\_backend&keyType=CLIENT](https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT)

You can also add your credentials to an existing key, using these values:

Package name:  
CE:38:5D:32:18:DF:57:7B:AF:1C:C2:72:C2:4F:B2:D8:92:C1:A9:F9

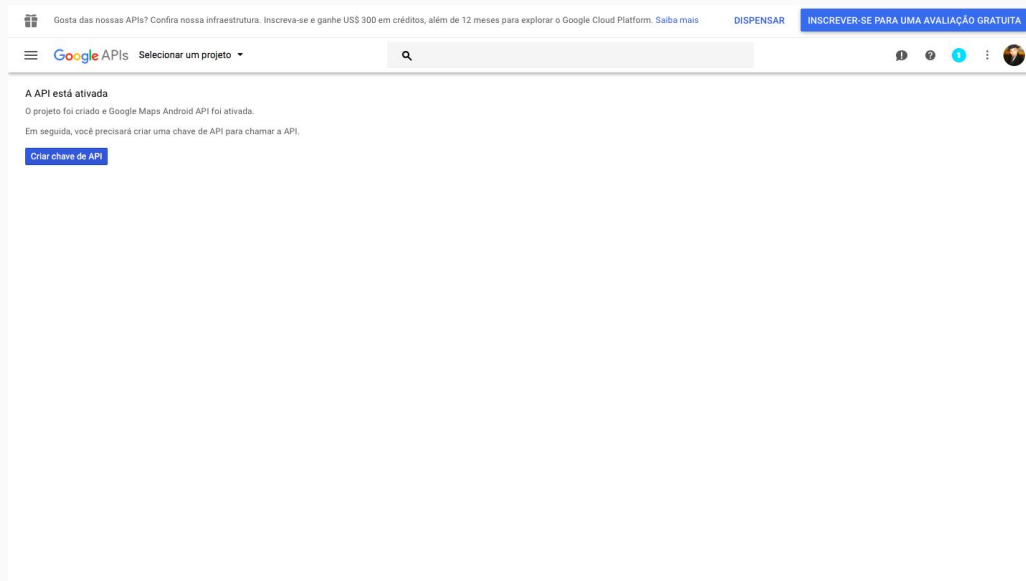
SHA-1 certificate fingerprint:  
CE:38:5D:32:18:DF:57:7B:AF:1C:C2:72:C2:4F:B2:D8:92:C1:A9:F9

Alternatively, follow the directions here:  
<https://developers.google.com/maps/documentation/android/start#get-key>

Once you have your key (it starts with "AIza"), replace the "google\_maps\_key" string in this file.

# Mapas

Para obter a chave, é preciso seguir as instruções apresentadas no link do google console, conforme foi demonstrado no slide anterior.





# Adicionando Pins

É fácil adicionar pins ao mapa. As interações com o mapa podem ser feitas através de latitude e longitude, e é bem simples adicionar um marcador.

Para adicionar um marcador basta iniciar um objeto LatLng e adicionar um MarkerOptions ao mapa.

```
/**
 * Manipulates the map once available.
 * This callback is triggered when the map is ready to be used.
 * This is where we can add markers or lines, add listeners or move the camera. In this case,
 * we just add a marker near Sydney, Australia.
 * If Google Play services is not installed on the device, the user will be prompted to install
 * it inside the SupportMapFragment. This method will only be triggered once the user has
 * installed Google Play services and returned to the app.
 */
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    // Add a marker in Sydney and move the camera
    LatLng sydney = new LatLng(-34, 151);
    mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
}
```

The end



Calma, ainda não!



# Material Design Guidelines

# Material Design Guidelines

Um ponto muito importante ao criar aplicativos é garantir uma boa experiência ao usuário. Para tanto, não só o aplicativo deve funcionar perfeitamente, quanto deve possuir comportamentos que os usuários já estejam acostumados, de acordo com cada plataforma que o aplicativo visa atender.

No nosso caso trabalhamos somente com o Android, portanto devemos direcionar a experiência do usuário para padrões comuns do sistema operacional

# Material Design Guidelines

Não seria produtivo apresentar aqui cada uma das guidelines informadas pela Google, uma vez que elas são muito bem explicadas em documentação própria para tal, a qual pode ser acessada [AQUI](#).

# Material Design Guidelines

É importante ressaltar que o uso destas guidelines não restringe a sua criatividade ao desenhar o seu aplicativo. Elas são somente uma série de boas práticas a se seguir quando desenvolve algo para uma ou outra plataforma, devendo ser observados seus princípios, não necessariamente sua forma.

Dúvidas?





Agora sim!

