



*Universidade Federal de Ouro Preto*  
*Departamento de Computação e Sistemas - DECSI*

## **Computação Móvel** **Fragments (Ref. Cap. 8)**

*Vicente Amorim*  
*vicente.amorim.ufop@gmail.com*  
*www.decom.ufop.br/vicente*



# Sumário

---

- \* Introdução
- \* Funcionamento
- \* Ciclo de vida
- \* Hello World Fragment & ActionBar + Tabs
- \* Migrando de Activity para Fragments
- \* Layouts Divididos Utilizando Fragments
- \* Fragments e a BackStack
- \* Salvando Estado de um Fragment



# Introdução

---



## Introdução

---

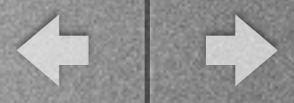
- Um **Fragment** é um componente de código reutilizável que possui autonomia para criar sua própria **view**, tratar eventos e gerenciar seu próprio ciclo de vida
- Representa um comportamento específico ou uma porção de interface de usuário de uma **Activity**.
- Introduzido a partir do **Android 3.0** (honeycomb).
- Pode ser compreendido como sendo um módulo de uma **Activity**.
- Um **Fragment** obrigatoriamente precisa estar dentro de uma **Activity**.



## Introdução

---

- O ciclo de vida de um **Fragment** é diretamente afetado pelo ciclo de vida da **Activity** no qual o mesmo está inserido.
- Se uma **Activity** está em seu estado *resumed*, cada **Fragment** interno pode estar em um estado diferente.
- Quando um **Fragment** é adicionado a uma **Activity**, o mesmo vive dentro de um **ViewGroup** interno.
- Assim como demais componentes, um **Fragment** pode ser adicionado a uma **Activity** via XML ou via API.

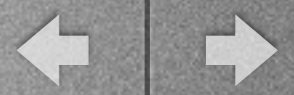


# Introdução

---

## - Design

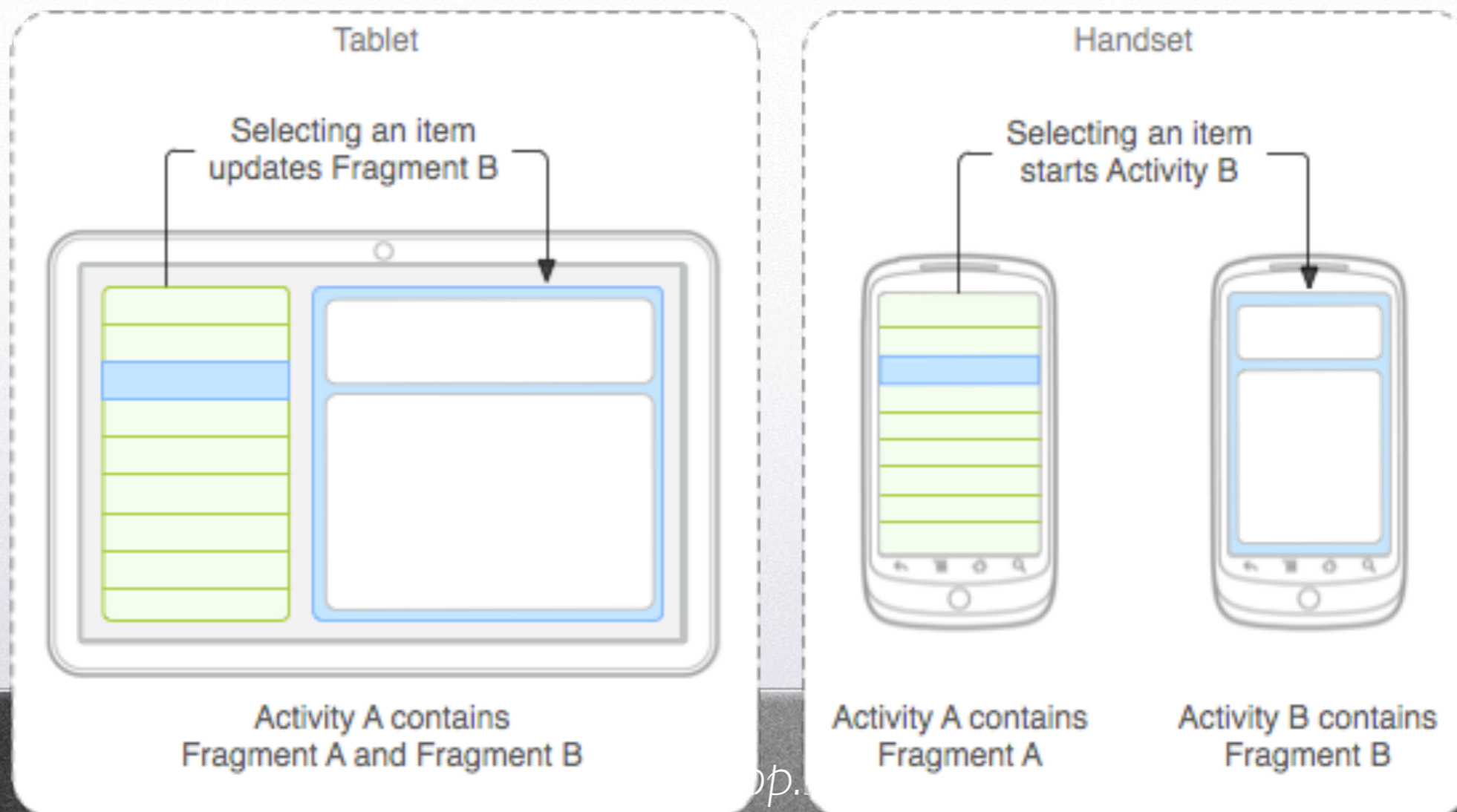
- ✓ **Fragments** foram introduzidos como forma de criar interfaces mais dinâmicas e flexíveis - principalmente em telas grandes (tablets).
- ✓ Permite a inserção de elementos visuais dinâmicos sem a necessidade de atividades (activities) complexas.
- ✓ É possível modificar a aparência de uma parte da **Activity** sem necessariamente mexer em todos os componentes visuais.
- ✓ Cada Fragment deve ser desenvolvido de forma modular, independente e reutilizável.



# Introdução

## - Design

✓ Diferentes *layouts* para um mesmo aplicativo (usando **Fragments** e somente com atividades - **Activities**):

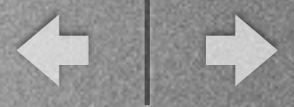




# Funcionamento

---





# Funcionamento

---

## - Introdução

- ✓ **Fragments** são mais que componentes “somente” para dividir a tela.
- ✓ Normalmente utilizados para mostrar simultaneamente diferentes informações de forma dinâmica na tela.
- ✓ Dependendo da organização interna, o código final da **Activity** pode ficar bem reduzido uma vez que tudo é delegado aos **Fragments**.
- ✓ O principal objetivo dos **Fragments** é criar módulos de software reutilizáveis.



# Funcionamento

---

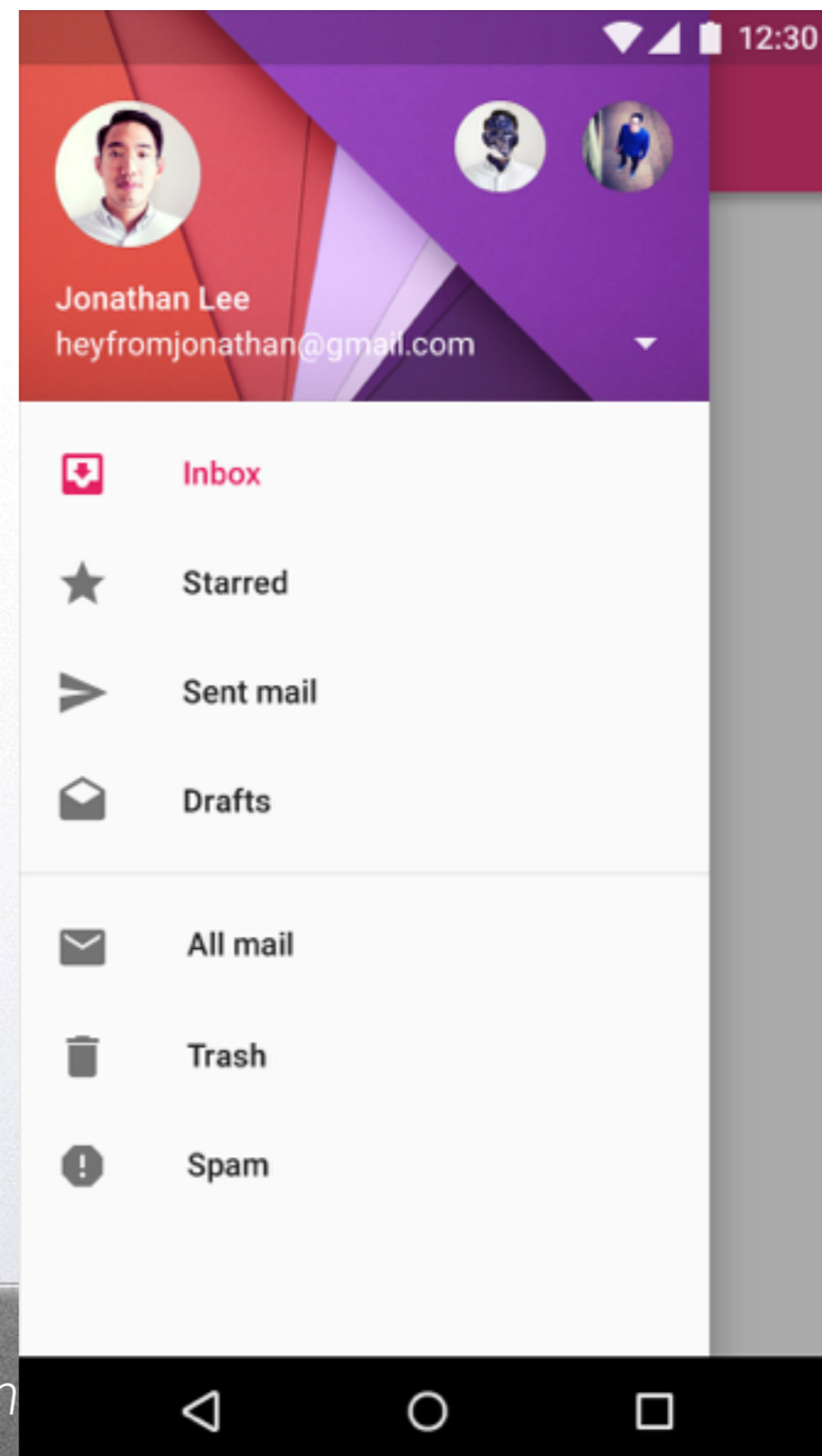
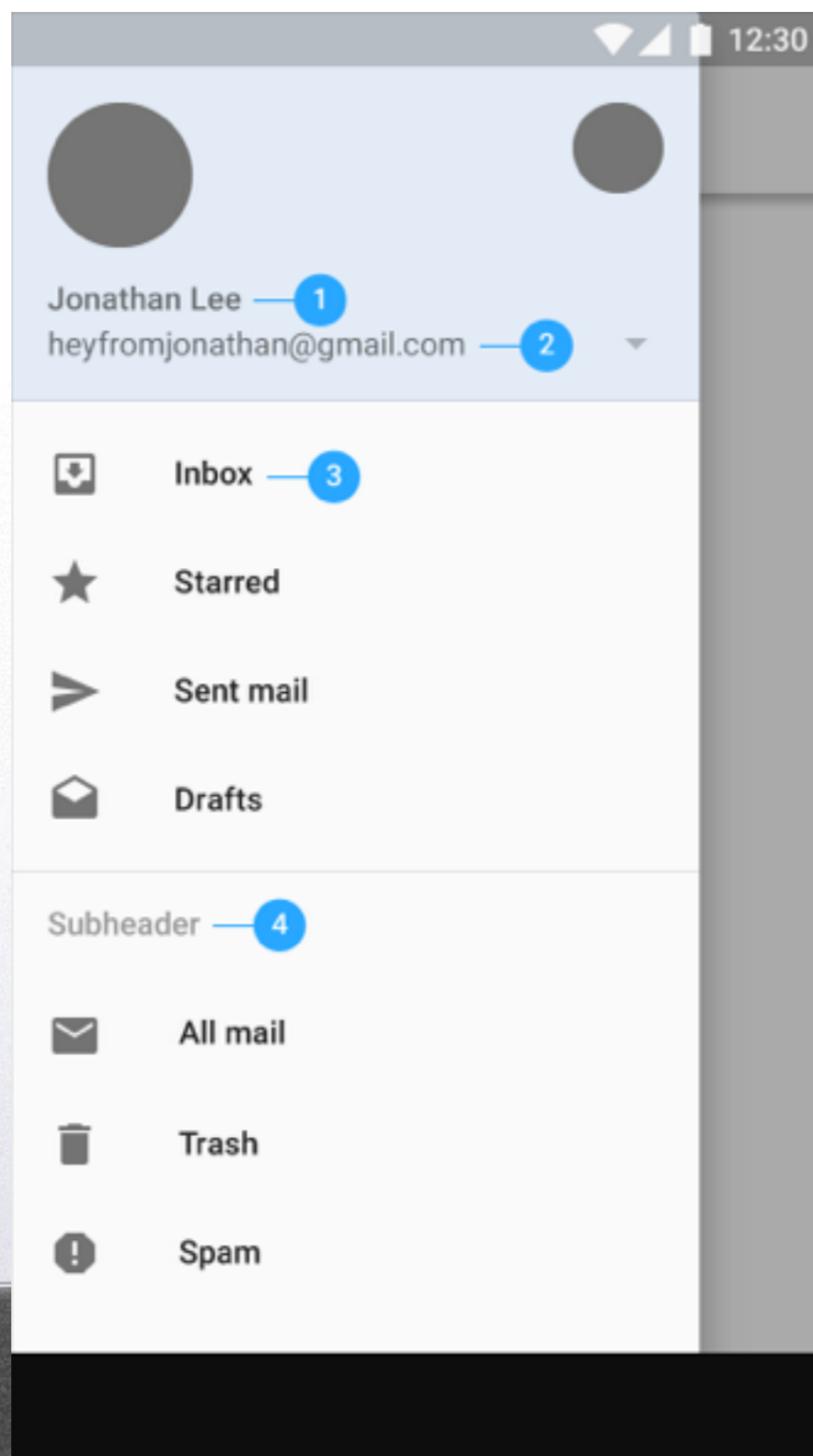
## - Introdução

- ✓ Ultimamente têm sido utilizados também na construção de aplicativos para *smartphones*.
- ✓ O aplicativo da loja Android (“Google Play”) atualmente faz uso dos **Fragments** em diversas partes.
- ✓ A modularização dos componentes visuais faz com que seja possível a utilização de várias telas a partir de somente uma **Activity**.
- ✓ Fragments integrados ao NavigationDrawer (menu lateral) tornam os aplicativos atuais muito mais funcionais.



# Funcionamento

## - Introdução





# Ciclo de vida

---



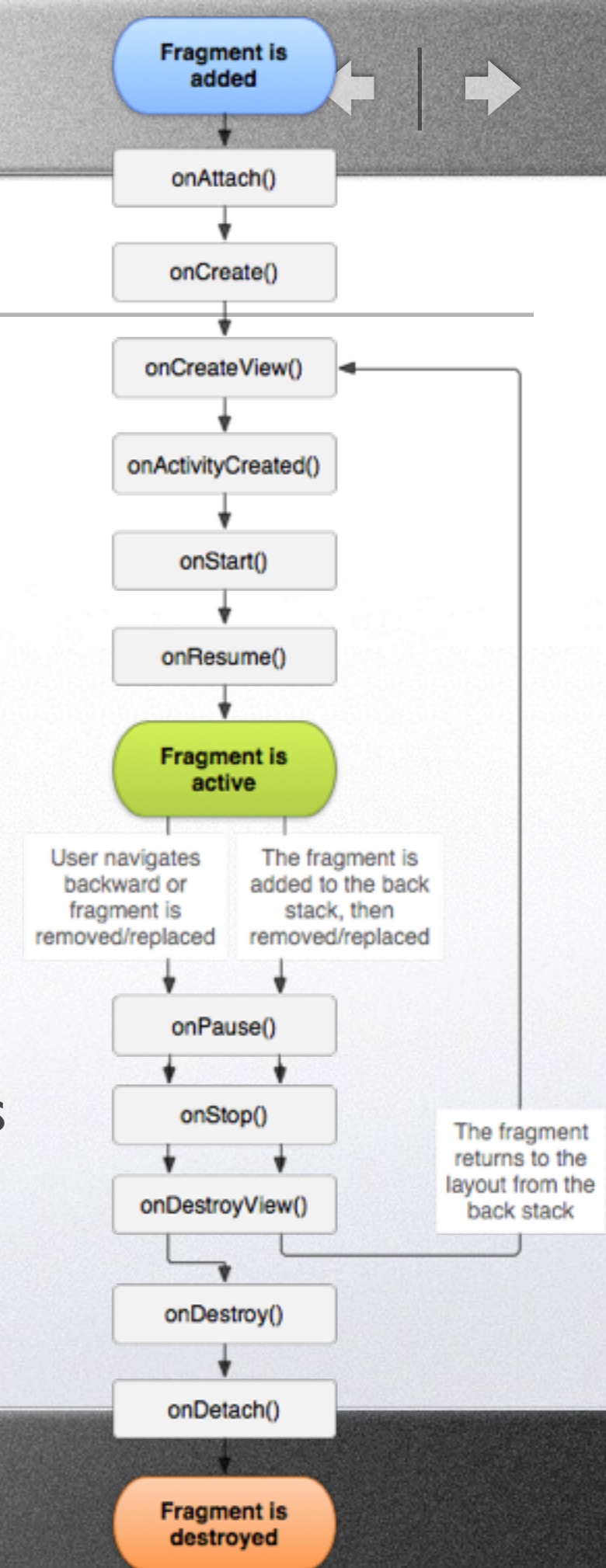
# Ciclo de vida

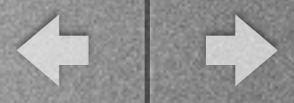
## - Introdução

✓ A criação de um **Fragment** envolve a implementação de uma subclasse de **Fragment**.

✓ A classe se parece muito com uma **Activity**, possuindo métodos de *callback* análogos: **onCreate()**, **onStart()**, **onPause()**, etc...

✓ Maioria dos aplicativos requer que pelo menos **onCreate()**, **onCreateView()** e **onPause()** sejam implementados - apesar de existirem outros.





# Ciclo de vida

---

## - Introdução

✓ Ciclo de vida de um **Fragment** é diretamente dependente do ciclo de vida de uma **Activity**:

◆ Se uma **Activity** é pausada, o sistema irá chamar o **onPause()** para todos os **Fragments** da mesma.

◆ Quando a **Activity** for resumida, o sistema irá chamar o **onResume()** para todos os **Fragments** da mesma.

✓ A maioria dos métodos do ciclo de vida de um **Fragment** é um espelho daqueles na **Activity**.



# Ciclo de vida

---

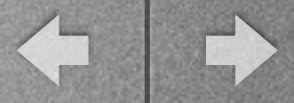
## - Introdução

✓ Quando uma **Activity** informa o seu layout através do método **setContentView(view)...**:

◆ **Fragments** são inflados e criados;

◆ Métodos **onAttach(activity)**, **onCreate()** e **onCreateView()** são chamados nos **Fragments**; e

◆ Quando **Activity** retorna do método **onCreate()**, **onActivityCreated()** é chamado nos **Fragments**.



# Ciclo de vida

---

## - Introdução

✓ Quando uma **Activity** é destruída, durante seu método **onDestroy()**...:

◆ Métodos **onDestroyView()**, **onDestroy()** e **onDetach()** são chamados em sequência.

◆ Tais métodos encerram e liberam recursos além de desassociar o **Fragment** da **Activity**.





# Ciclo de vida

## - Introdução

### ✓ Principais métodos do ciclo de vida dos **Fragments**:

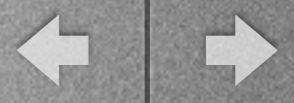
Método	Descrição
<code>onAttach(Activity)</code>	Chamado logo após o Fragment ser associado com a Activity — logo após a Activity inflar o layout. Pode ser invocada também após o Fragment ser adicionado dinamicamente através de um <code>FragmentManager</code> .
<code>onCreate(bundle)</code>	Chamado apenas uma vez e quando o Fragment está sendo criado. Funcionamento análogo ao da Activity.
<code>onCreateView(inflater, viewGroup, bundle)</code>	Método invocado quando o Fragment precisa criar a View a ser exibida. Somente após a execução do mesmo é possível se chamar o <code>getView()</code> do Fragment.
<code>onActivityCreated(bundle)</code>	Chamado logo após o método <code>onCreate()</code> ter sido finalizado. Pode ser utilizado como marca para consultas a <code>WebServices</code> ou <code>BDs</code> externos.
<code>onDestroyView()</code>	Chamado quando a view do Fragment foi removida. Após tal método o <code>getView()</code> retornará <code>null</code> .
<code>onDestroy()</code>	Chamado para indicar que o Fragment será destruído.
<code>onDetach()</code>	Chamado quando o Fragment é desassociado da Activity.



# Hello World Fragment

---





# Hello World Fragment

---

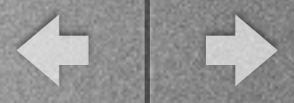
## - Introdução

✓ Exemplo: [AppNum55](#) - activity\_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent" android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.cea436.appnum55.MainActivity">
```

```
<fragment android:id="@+id/frag1"
    android:layout_width="match_parent" android:layout_height="match_parent"
    class="com.cea436.appnum55.Fragment1" tools:layout="@layout/fragment_fragment1"/>
```

```
</LinearLayout>
```



# Hello World Fragment

---

## - Introdução

✓ Exemplo: [AppNum55](#) - fragment\_fragment1.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.cea436.appnum55.Fragment1">

    <TextView
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:gravity="center" android:text="@string/hello_blank_fragment" />

</LinearLayout>
```



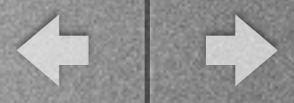
# Hello World Fragment

---

## - Introdução

✓ Exemplo: [AppNum55](#) - Fragment1.java

```
public class Fragment1 extends Fragment {  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                             Bundle savedInstanceState) {  
        View view = inflater.inflate(R.layout.fragment_fragment1, container, false);  
        //Lógica do fragment...  
        return view;  
    }  
}
```



# Hello World Fragment

---

## - Introdução

✓ Exemplo: [AppNum55](#) - MainActivity.java

```
public class MainActivity extends FragmentActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        FragmentManager fm = getSupportFragmentManager();  
    }  
}
```



# Hello World Fragment

---

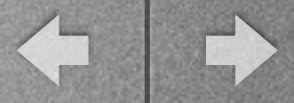
- Introdução

✓ Exemplo: [AppNum55](#)

3G 2:44

Hello Fragment 1!!





# Hello World Fragment

---

## - Introdução

- ✓ Assim como outros componentes **Android**, os **Fragments** também podem ser criados via API.
- ✓ Cria-se somente o layout relacionado o qual irá conter o **Fragment**.
- ✓ Demais componentes podem ser adicionados dinamicamente ao **Fragment** após sua criação e inclusão no layout.





# Hello World Fragment

---

## - Introdução

✓ Exemplo: [AppNum56](#) - MainActivity.java

```
public class MainActivity extends FragmentActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        if (savedInstanceState == null) {  
            FragmentManager fm = getSupportFragmentManager();  
            android.support.v4.app.FragmentTransaction ft = fm.beginTransaction();  
            Fragment frag1 = new Fragment();  
            ft.add(R.id.layoutFrag, frag1, "Fragment 1");  
            ft.commit();  
        }  
    }  
}
```



# Hello World Fragment

---

- Introdução

✓ Exemplo: [AppNum56](#)

3G 5:06





# ActionBar + Tabs

---



## ActionBar + Tabs

---

### - Introdução

- ✓ Anteriormente foram feitos exemplos de códigos que utilizavam **ActionBar** juntamente com as **Tabs**.
- ✓ Ainda, é possível integrar tais soluções com ao conceito de **Fragments** para gerar soluções mais “elegantes”.
- ✓ Independente da forma de exibição das opções (Tabs ou NavigationDrawer), quando uma opções é selecionada, o conteúdo da tela deve ser alterado sem necessidade de mudança da Activity.



# ActionBar + Tabs

---

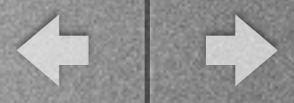
## - Introdução

✓ Exemplo: [AppNum57](#) - my\_fragment.xml / my\_fragment2.xml

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

    <TextView android:id="@+id/tv1"
        android:layout_width="match_parent" android:layout_height="match_parent"
        android:layout_gravity="center" android:text="@string/fragString"/>

</LinearLayout>
```



# ActionBar + Tabs

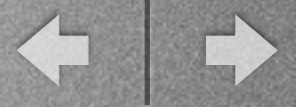
---

## - Introdução

✓ Exemplo: [AppNum57](#) - activity\_main.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:id="@+id/layoutFrag"
  tools:context="com.cea436.appnum57.MainActivity">

</FrameLayout>
```



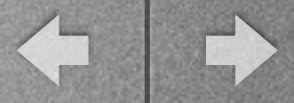
# ActionBar + Tabs

---

## - Introdução

✓ Exemplo: [AppNum57](#) - MyFragment.java / MyFragment2.java

```
public class MyFragment extends android.support.v4.app.Fragment {  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle  
savedInstanceState) {  
        View view = inflater.inflate(R.layout.my_fragment, container, false);  
        return view;  
    }  
}
```



# ActionBar + Tabs

## - Introdução

### ✓ Exemplo: [AppNum57](#) - MyTabListener.java

```
public class MyTabListener implements android.support.v7.app.ActionBar.TabListener {
    private Context context;
    private android.support.v4.app.Fragment frag;

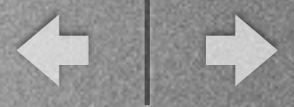
    public MyTabListener(Context context, android.support.v4.app.Fragment frag) {
        this.context = context;
        this.frag = frag;
    }

    @Override
    public void onTabSelected(android.support.v7.app.ActionBar.Tab tab,
        android.support.v4.app.FragmentTransaction ft) {
        //Troca o fragment dinamicamente ao clicar na tab
        ft.replace(R.id.layoutFrag, this.frag, null);
        Toast.makeText(context, "Tab Seleccionada!!", Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onTabUnselected(...) { }

    @Override
    public void onTabReselected(...) { }
}
```





# ActionBar + Tabs

## - Introdução

### ✓ Exemplo: [AppNum57](#) - MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        android.support.v7.app.ActionBar actionBar = getSupportActionBar();  
        actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);  
  
        //Tab 1  
        android.support.v7.app.ActionBar.Tab tab1 = actionBar.newTab().setText("Frag 1");  
        tab1.setTabListener(new MyTabListener(this, new MyFragment()));  
        actionBar.addTab(tab1);  
  
        //Tab 2  
        android.support.v7.app.ActionBar.Tab tab2 = actionBar.newTab().setText("Frag 2");  
        tab2.setTabListener(new MyTabListener(this, new MyFragment2()));  
        actionBar.addTab(tab2);  
    }  
}
```



# ActionBar + Tabs

- Introdução

✓ Exemplo: [AppNum57](#)

AppNum57

FRAG 1

FRAG 2

Meu fragment 1!!

3G 5:59



# Migrando de Activity para Fragments

---



# Migrando de Activity para Fragments

---

## - Introdução

- ✓ Utilizando **Fragments** é possível refatorar aplicativos de *smartphone* para *tablets*.
- ✓ Possibilita reaproveitar código além de economizar tempo de desenvolvimento e debug.
- ✓ **Importante:** Lembrar que em telas maiores a disposição dos elementos deve ser reconsiderada.
- ✓ No próximo exemplo será demonstrado como migrar um aplicativo previamente desenvolvido em *smartphones* para Tablets.



# Migrando de Activity para Fragments

---

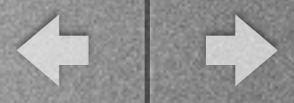
## - Introdução

✓ Exemplo [AppNum58](#) - Parte I - Aplicativo p/ *smartphones*

◆ adapter\_planeta.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="?android:attr/listPreferredItemHeight"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:gravity="center_vertical"
        android:layout_height="match_parent" />
</LinearLayout>
```



# Migrando de Activity para Fragments

## - Introdução

✓ Exemplo [AppNum58](#) - Parte I - Aplicativo p/ *smartphones*

◆ activity\_planeta.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="com.cea436.appnum58.PlanetaActivity">

    <TextView
        android:id="@+id/text"
        android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```



# Migrando de Activity para Fragments

---

## - Introdução

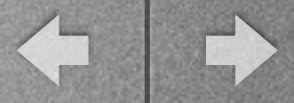
✓ Exemplo [AppNum58](#) - Parte I - Aplicativo p/ *smartphones*

◆ activity\_main.xml

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

    <ListView
        android:id="@+id/listview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp" />

</RelativeLayout>
```



# Migrando de Activity para Fragments

---

## - Introdução

✓ Exemplo [AppNum58](#) - Parte I - Aplicativo p/ *smartphones*

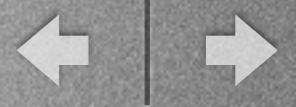
### ◆ PlanetaAdapter.java

```
public class PlanetaAdapter extends BaseAdapter {
    private String[] planetas = new String[]{"Mercúrio", "Vênus", "Terra", "Marte", "Júpiter",
        "Saturno", "Urano", "Netuno", "Plutão"};
    private Context context;

    public PlanetaAdapter(Context context) {
        super();
        this.context = context;
    }

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





# Migrando de Activity para Fragments

## - Introdução

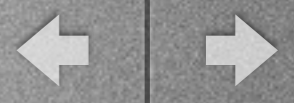
✓ Exemplo [AppNum58](#) - Parte I - Aplicativo p/ *smartphones*

### ◆ PlanetaAdapter.java

```
...
@Override
public Object getItem(int position) {
    return planetas[position];
}

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

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    String planeta = planetas[position];
    View view = LayoutInflater.from(context).inflate(R.layout.adapter_planeta, parent, false);
    TextView t = (TextView) view.findViewById(R.id.text);
    t.setText(planeta);
    return view;
}
```



# Migrando de Activity para Fragments

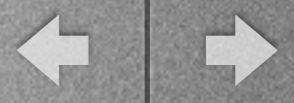
---

## - Introdução

✓ Exemplo [AppNum58](#) - Parte I - Aplicativo p/ *smartphones*

◆ PlanetaActivity.java

```
public class PlanetaActivity extends ActionBarActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_planeta);  
  
        String planeta = getIntent().getStringExtra("planeta");  
  
        TextView text = (TextView) findViewById(R.id.text);  
        text.setText(planeta);  
  
        getSupportActionBar().setTitle(planeta);  
    }  
}
```



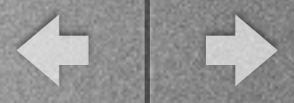
# Migrando de Activity para Fragments

## - Introdução

✓ Exemplo [AppNum58](#) - Parte I - Aplicativo p/ *smartphones*

◆ MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        // ListView  
        ListView listView = (ListView) findViewById(R.id.listview);  
        listView.setAdapter(new PlanetaAdapter(this));  
        listView.setOnItemClickListener(onItemClickPlaneta());  
    }  
    ...  
}
```



# Migrando de Activity para Fragments

## - Introdução

✓ Exemplo [AppNum58](#) - Parte I - Aplicativo p/ *smartphones*

### ◆ MainActivity.java

```
private AdapterView.OnItemClickListener onItemClickPlaneta() {
    return new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
            PlanetaAdapter adapter = (PlanetaAdapter) parent.getAdapter();
            String planeta = (String) adapter.getItem(position);
            Toast.makeText(getBaseContext(), "Planeta: " + planeta,
                Toast.LENGTH_SHORT).show();

            Intent intent = new Intent(getBaseContext(), PlanetaActivity.class);
            intent.putExtra("planeta", planeta);
            startActivity(intent);
        }
    };
}
```

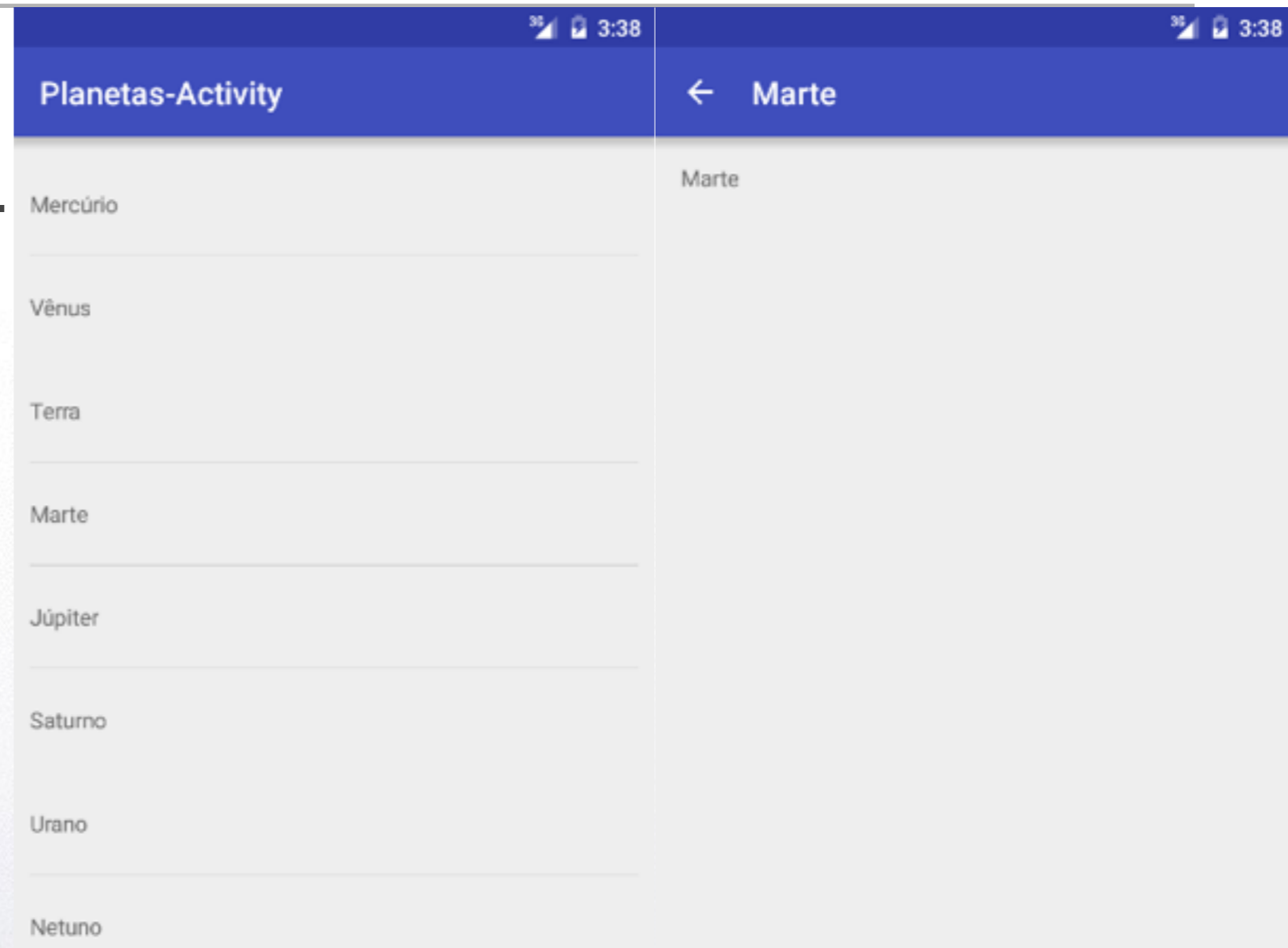




# Migrando de Activity para Fragments

## - Introdução

✓ Exemplo [AppNum58](#) -  
Parte I - Aplicativo p/  
*smartphones*





# Migrando de Activity para Fragments

---

## - Introdução

- ✓ O “Exemplo [AppNum58](#) Parte I” utiliza somente atividades para tratamento das entradas e exibição dos dados.
- ✓ A classe [MainActivity](#) exibe a lista dos planetas:
  - ◆ Quando um planeta é selecionado, uma intent é gerada e passada para a segunda [Activity](#); e
  - ◆ Segunda [Activity](#) recebe o dado informado e exibe as informações relacionadas ao planeta selecionado.
- ✓ Próximo exemplo passará toda a lógica para dois [Fragments](#).



# Migrando de Activity para Fragments

---

## - Introdução

### ✓ Exemplo [AppNum58](#) - Parte 2

#### ◆ fragment\_planetas.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent" android:layout_height="match_parent"
    android:orientation="vertical" android:padding="16dp"
    tools:context="com.cea436.appnum58pt2.PlanetasFragment">

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

</LinearLayout>
```



# Migrando de Activity para Fragments

---

## - Introdução

### ✓ Exemplo [AppNum58](#) - Parte 2

#### ◆ fragment\_planeta.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.cea436.appnum58pt2.PlanetaFragment">

    <TextView android:id="@+id/text"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_gravity="center" />
</FrameLayout>
```





# Migrando de Activity para Fragments

---

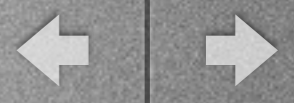
## - Introdução

### ✓ Exemplo [AppNum58](#) - Parte 2

#### ◆ adapter\_planeta.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="?android:attr/listPreferredItemHeight"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:gravity="center_vertical"
        android:layout_height="match_parent" />
</LinearLayout>
```



# Migrando de Activity para Fragments

## - Introdução

### ✓ Exemplo [AppNum58](#) - Parte 2

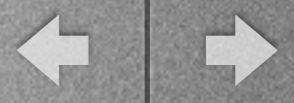
#### ◆ activity\_planeta.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="com.cea436.appnum58pt2.PlanetaActivity">
```

```
<fragment android:id="@+id/PlanetaFragment"
    android:layout_width="match_parent" android:layout_height="match_parent"
    class="com.cea436.appnum58pt2.PlanetaFragment" tools:layout="@layout/fragment_planeta"/>
```

```
</RelativeLayout>
```





# Migrando de Activity para Fragments

---

## - Introdução

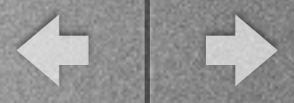
✓ Exemplo [AppNum58](#) - Parte 2

◆ main\_activity.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools">
```

```
<fragment android:id="@+id/planetasFragment"
    android:layout_width="match_parent" android:layout_height="match_parent"
    class="com.cea436.appnum58pt2.PlanetasFragment" tools:layout="@layout/fragment_planetas"/>
```

```
</RelativeLayout>
```



# Migrando de Activity para Fragments

## - Introdução

### ✓ Exemplo [AppNum58](#) - Parte 2

#### ◆ PlanetasFragment.java

```
public class PlanetasFragment extends Fragment {

    public PlanetasFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_planetas, container, false);
        //ListView relacionado
        ListView listview = (ListView) view.findViewById(R.id.listview);
        listview.setAdapter(new PlanetaAdapter(getActivity()));
        listview.setOnItemClickListener(onItemClickPlaneta());
        return view;
    }
    ...
}
```



# Migrando de Activity para Fragments

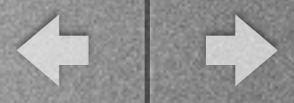
## - Introdução

### ✓ Exemplo [AppNum58](#) - Parte 2

#### ◆ PlanetasFragment.java

```
...
private AdapterView.OnItemClickListener onItemClickPlaneta() {
    return new AdapterView.OnItemClickListener() {

        @Override
        public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
            PlanetaAdapter adapter = (PlanetaAdapter) adapterView.getAdapter();
            String planeta = (String) adapter.getItem(i);
            //O context é a activity, então pode-se utilizar o método getActivity()
            Toast.makeText(getActivity(), "Planeta: " + planeta, Toast.LENGTH_SHORT).show();
            //Navegacao de tela continua sendo feita pela activity
            Intent intent = new Intent(getActivity(), PlanetaActivity.class);
            intent.putExtra("planeta", planeta);
            startActivity(intent);
        }
    };
}
```



# Migrando de Activity para Fragments

## - Introdução

### ✓ Exemplo [AppNum58](#) - Parte 2

#### ◆ PlanetaFragment.java

```
public class PlanetaFragment extends Fragment {  
  
    public PlanetaFragment() {  
    }  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                             Bundle savedInstanceState) {  
        return inflater.inflate(R.layout.fragment_planeta, container, false);  
    }  
  
    public void setPlaneta(String planeta) {  
        TextView tv = (TextView)getView().findViewById(R.id.text);  
        tv.setText("Planeta: "+planeta);  
    }  
}
```



# Migrando de Activity para Fragments

---

## - Introdução

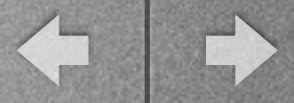
### ✓ Exemplo [AppNum58](#) - Parte 2

#### ◆ PlanetaAdapter.java

```
public class PlanetaAdapter extends BaseAdapter {
    private String[] planetas = new String[]{"Mercúrio", "Vênus", "Terra", "Marte", "Júpiter",
        "Saturno", "Urano", "Netuno", "Plutão"};
    private Context context;

    public PlanetaAdapter(Context context) {
        super();
        this.context = context;
    }

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



# Migrando de Activity para Fragments

## - Introdução

### ✓ Exemplo [AppNum58](#) - Parte 2

#### ◆ PlanetaAdapter.java

```
...
@Override
public Object getItem(int position) {
    return planetas[position];
}

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

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    String planeta = planetas[position];
    View view = LayoutInflater.from(context).inflate(R.layout.adapter_planeta, parent, false);
    TextView t = (TextView) view.findViewById(R.id.text);
    t.setText(planeta);
    return view;
}
}
```





# Migrando de Activity para Fragments

## - Introdução

### ✓ Exemplo [AppNum58](#) - Parte 2

#### ◆ PlanetaActivity.java

```
public class PlanetaActivity extends ActionBarActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_planeta);  
  
        String planeta = getIntent().getStringExtra("planeta");  
        PlanetaFragment pf =  
(PlanetaFragment) getSupportFragmentManager().findFragmentById(R.id.PlanetaFragment);  
        pf.setPlaneta(planeta);  
        getSupportActionBar().setTitle(planeta);  
    }  
}
```



# Migrando de Activity para Fragments

---

## - Introdução

✓ Exemplo [AppNum58](#) - Parte 2

◆ MainActivity.java

```
public class MainActivity extends FragmentActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

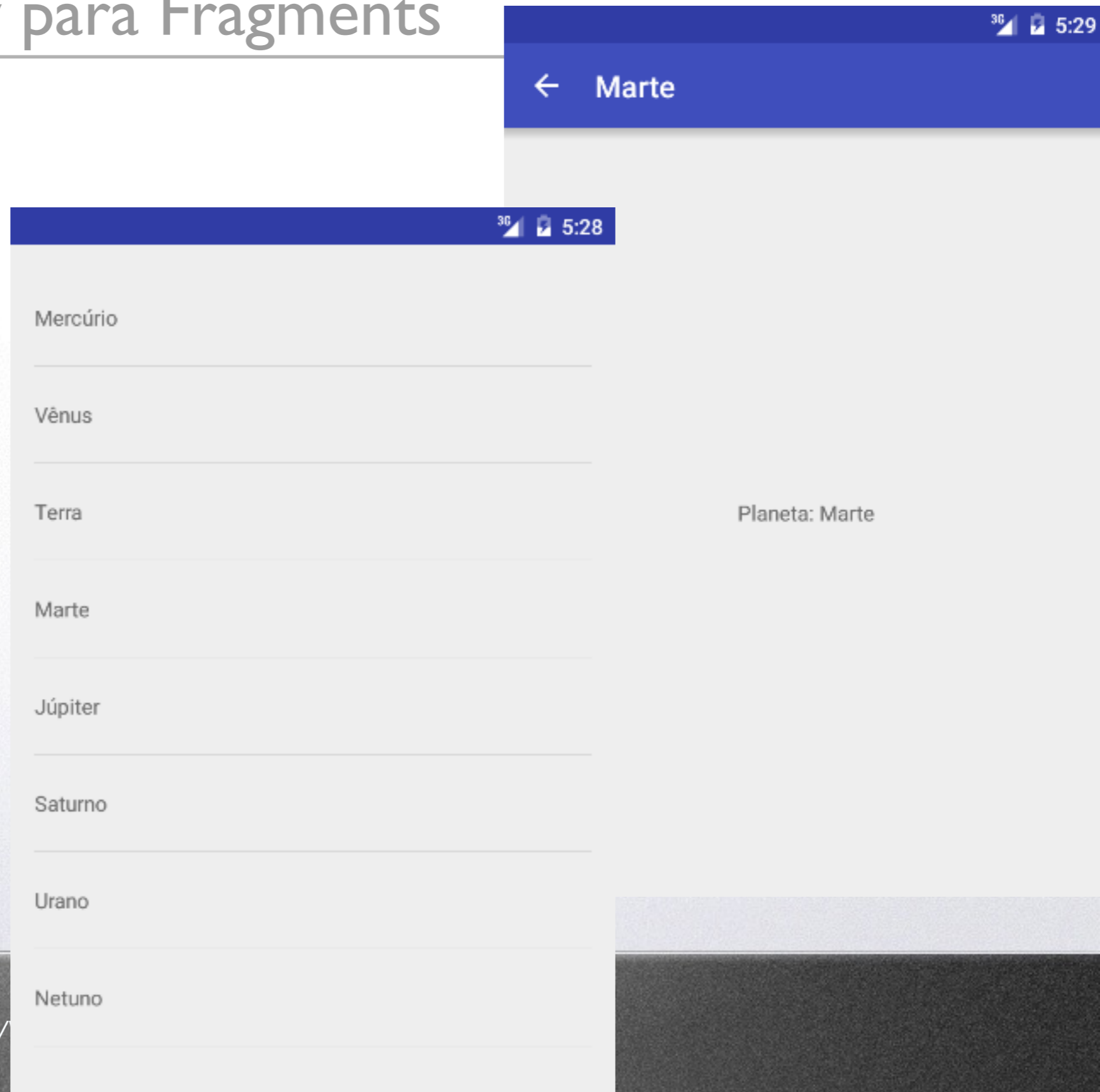


# Migrando de Activity para Fragments

## - Introdução

✓ Exemplo

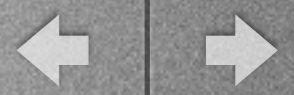
[AppNum58](#) - Parte 2





# Layouts Divididos Utilizando Fragments

---



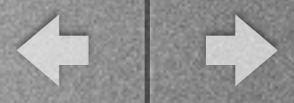
# Layouts Divididos Utilizando Fragments

## - Introdução

✓ O exemplo AppNum58 (pts. 1 e 2) será agora adaptado para o layout em duas diferentes partes.

✓ Para criar um layout específico para tablets é possível utilizar a seguinte convenção de pastas:

Pasta	Descrição
<i>/res/layout-large</i> <i>/res/layout-sw600dp</i>	Layouts para Tablets de 7”.
<i>/res/layout-xlarge</i> <i>/res/layout-sw720dp</i>	Layouts para Tablets de 10”.
<i>/res/layout-large-land</i> <i>/res/layout-xlarge-land</i>	Layouts “landscape” para Tablets de 10”.



# Layouts Divididos Utilizando Fragments

---

## - Introdução

### ✓ Exemplo [AppNum59](#)

- ◆ Utilizado um emulador de **Tablet de 7"** (600 x 1024 mdpi).
- ◆ Arquivo [activity\\_main.xml](#) foi mantido e um novo adicionado à pasta ["/res/layout-large-land"](#).
- ◆ Organização dos **Fragments** na tela é feita através do próprio layout principal.



# Layouts Divididos Utilizando Fragments

## - Introdução

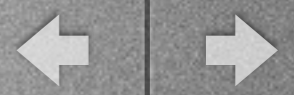
✓ Exemplo **AppNum59** - /res/layout-large-land/activity\_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    xmlns:tools="http://schemas.android.com/tools">
```

```
<fragment android:id="@+id/planetasFragment"
    android:layout_width="0dp" android:layout_height="match_parent"
    android:layout_weight="1"
    class="com.cea436.appnum59.PlanetasFragment" tools:layout="@layout/fragment_planetas"/>
```

```
<fragment android:id="@+id/PlanetaFragment"
    android:layout_width="0dp" android:layout_height="match_parent"
    android:layout_weight="1"
    class="com.cea436.appnum59.PlanetaFragment" tools:layout="@layout/fragment_planeta"/>
```

```
</LinearLayout>
```



# Layouts Divididos Utilizando Fragments

## - Introdução

### ✓ Exemplo [AppNum59](#) - PlanetasFragment.java

...

```
@Override
public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
    PlanetaAdapter adapter = (PlanetaAdapter) adapterView.getAdapter();
    String planeta = (String) adapter.getItem(i);
    //O context é a activity, então pode-se utilizar o método getActivity()
    Toast.makeText(getActivity(), "Planeta: " + planeta, Toast.LENGTH_SHORT).show();
    //Navegacao de tela continua sendo feita pela activity
    PlanetaFragment pf = (PlanetaFragment)
getFragmentManager().findFragmentById(R.id.PlanetaFragment);
    boolean dualLayout = pf != null;
    if (dualLayout) {
        //Só atualiza o frame da direita se é dual layout
        pf.setPlaneta(planeta);
    } else {
        Intent intent = new Intent(getActivity(), PlanetaActivity.class);
        intent.putExtra("planeta", planeta);
        startActivity(intent);
    }
}
```

...



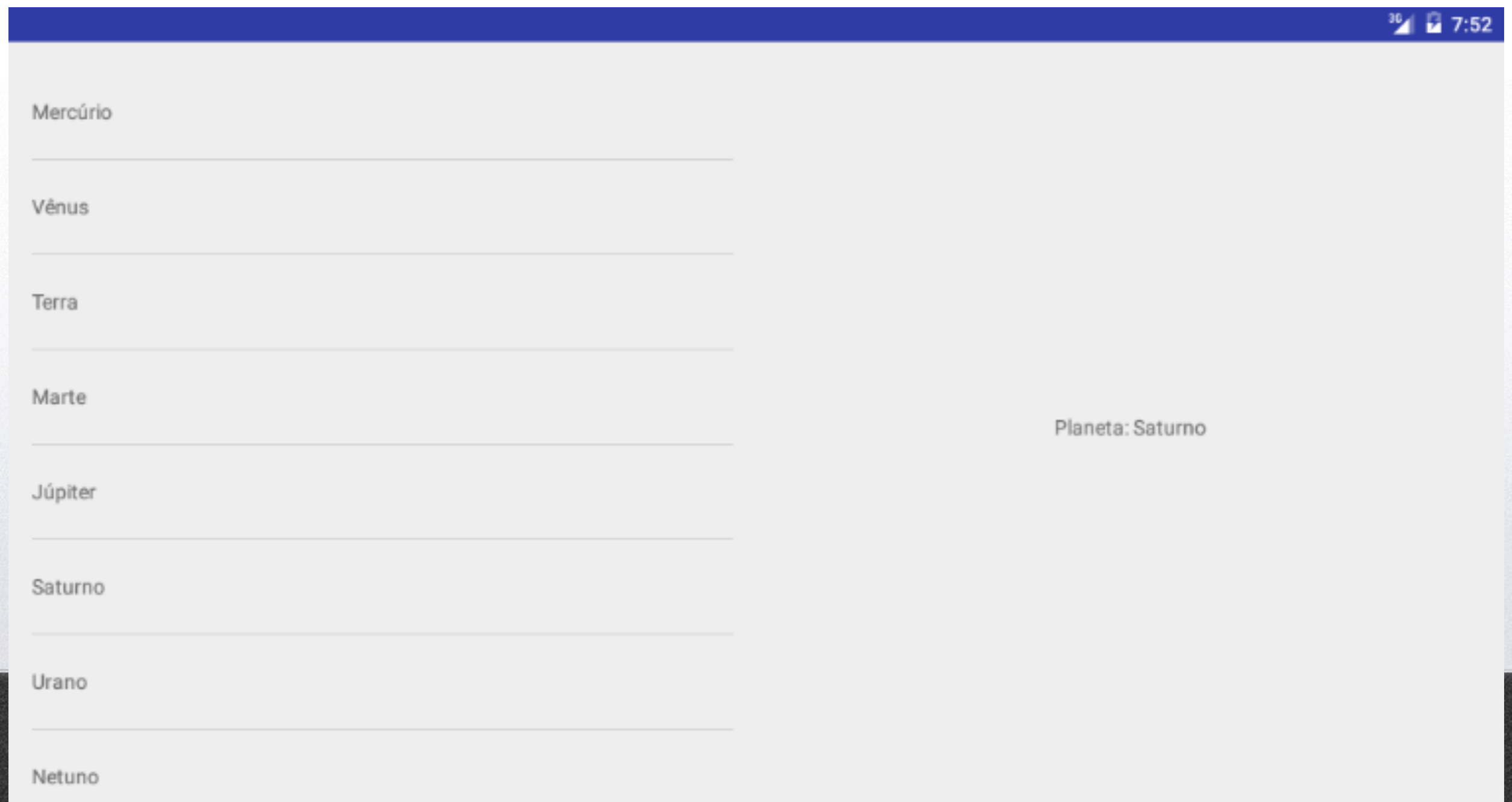


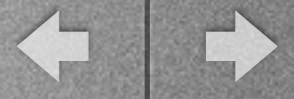


# Layouts Divididos Utilizando Fragments

## - Introdução

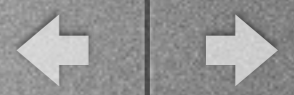
✓ Exemplo [AppNum59](#)





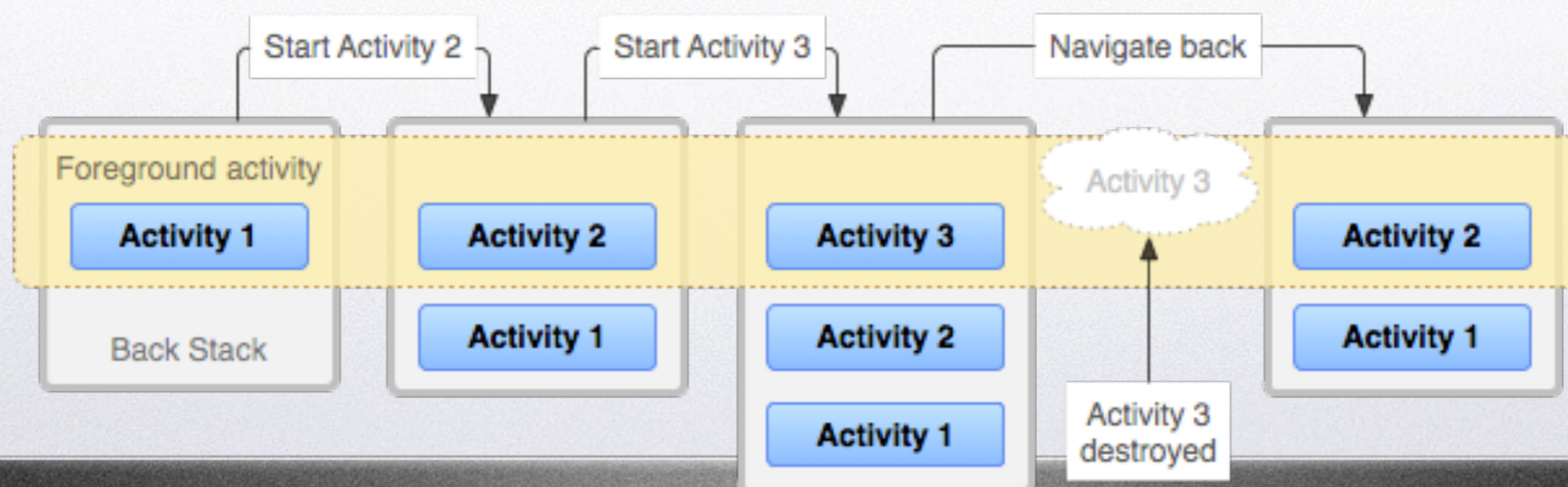
# Fragments e a BackStack

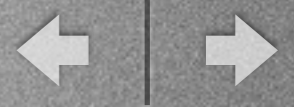
---



## Fragments e a BackStack

- É a estrutura que representa a organização das atividades em memória.
- Cada vez que um atividade é criada, a mesma é adicionada no topo da *BackStack*.
- Cada atividade removida é analogamente removida do topo da *BackStack*.





## Fragments e a BackStack

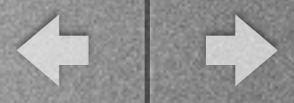
---

- Os Fragments também podem ser gerenciados via *BackStack*:
  - ✓ Na classe **FragmentTransaction**:
    - ◆ **addToBackStack(tag)**: Adiciona um Fragment na BackStack — ao se clicar no botão “voltar”, a operação é desfeita.\*
    - ◆ **popBackStack()**: Desempilha o último Fragment da BackStack — funciona como o próprio botão “voltar”.
    - ◆ Toda operação executada através da classe **FragmentTransaction** deve ser — ao final — seguido de uma chamada ao método **commit()**.
  - \* Parâmetro “**tag**” é o identificador da transação. Caso desnecessário, pode ser passado “**null**”.



# Salvando Estado de um Fragment

---



## Salvando Estado de um Fragment

---

- É frequente a necessidade de salvamento do estado de uma aplicação durante sua execução:
  - ✓ Ex.: Usuário pode girar a tela durante o download de dados vindos de um WebService.
  - ✓ **Android** destrói a **Activity** (e seus **Fragments**) para depois recriá-los.
  - ✓ O estado dos dados precisa ser salvo e posteriormente recuperado.
  - ✓ Plataforma facilita muito tal serviço através da utilização de mecanismo de manutenção do estado.



# Salvando Estado de um Fragment

- Detalhamento:

✓ Método 1:

```
public class Fragment1 extends android.support.v4.app.Fragment {  
  
    private int count;  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                             Bundle savedInstanceState) {  
  
        View view = inflater.inflate(R.layout.fragment_1, container, false);  
  
        ...  
  
        // Recupera o estado da variável  
        if (savedInstanceState != null) {  
            count = savedInstanceState.getInt("count");  
        }  
  
        ...  
    }  
  
    ...  
    @Override  
    public void onSaveInstanceState(Bundle outState) {  
        super.onSaveInstanceState(outState);  
        outState.putInt("count", count);  
    }  
  
    ...  
}
```



# Salvando Estado de um Fragment

---

## - Detalhamento:

### ✓ Método 2:

```
public class Fragment1 extends android.support.v4.app.Fragment {  
    private int count;  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        View view = inflater.inflate(R.layout.fragment_1, container, false);  
  
        ...  
        // Ou apenas faz isso.  
        setRetainInstance(true);  
        ...  
    }  
}
```





## Salvando Estado de um Fragment

---

### - Detalhamento:

- ✓ No **Método 1** o desenvolvedor é responsável por organizar e gerenciar quais dados serão armazenados durante a destruição/recriação da Activity/Fragments.
- ✓ No **Método 2** todo o objeto é colocado em memória sem distinção de informações.
- ✓ O primeiro é mais eficiente em termos de memória e o segundo é mais simples de ser implementado.