

Programação Para Dispositivos Móveis I

FRAGMENTS

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis

Ricardo Barbosa , rmb@estg.ipp.pt

Carlos Aldeias, cfpa@estg.ipp.pt

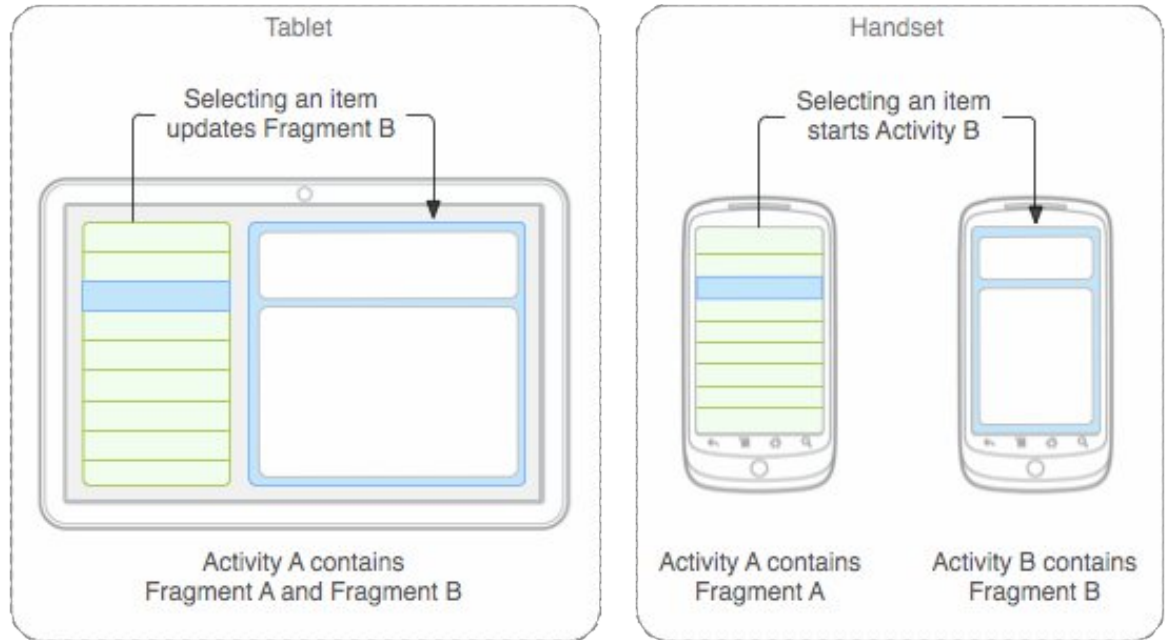
Índice

- Fragments;
- Ciclo de Vida dos Fragments;
- Criar um Fragment;
- Substituir Fragments num FrameLayout;
- Comunicação Entre Fragments;
- Tablet e Smartphone;
- Leitura Adicional.

Fragments

Um **comportamento** ou uma **parte da interface gráfica** de uma Activity;

Podem ser **utilizados mais do que um** por Activity e podem ser utilizados em **mais do que uma** Activity.



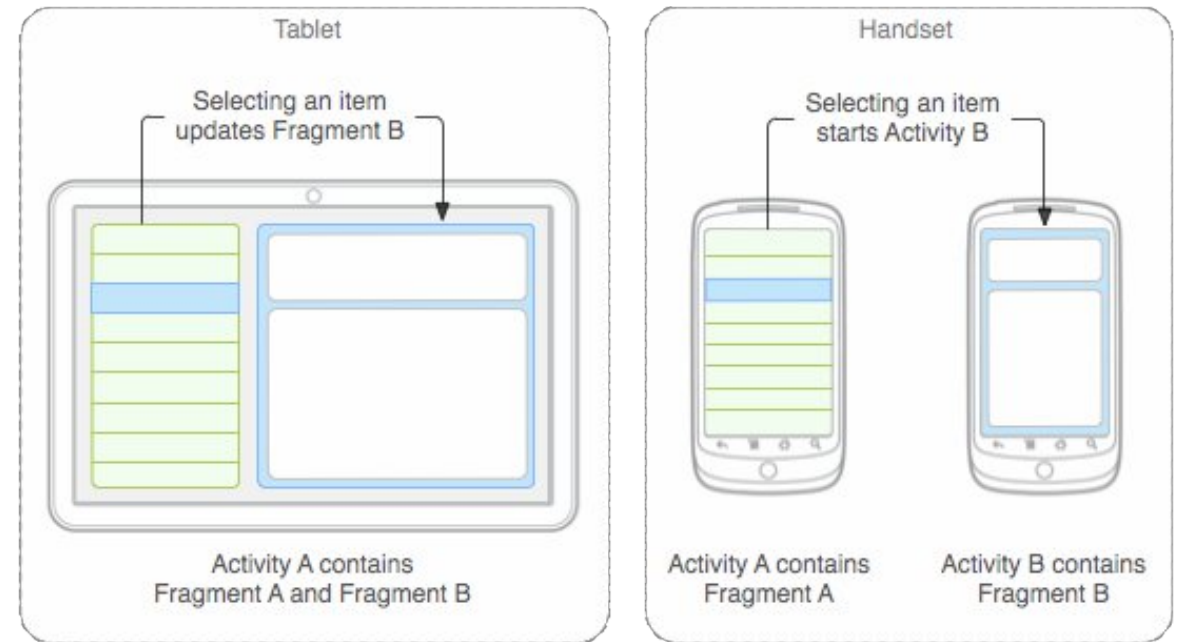
Fragments

Definição

É uma combinação de um layout xml, e uma classe (de forma semelhante a uma Activity);

Fazem o encapsulamento de views e lógica, para que sejam de fácil reutilização em Activities;

São componentes que contêm views, eventos e lógica;



Fragments

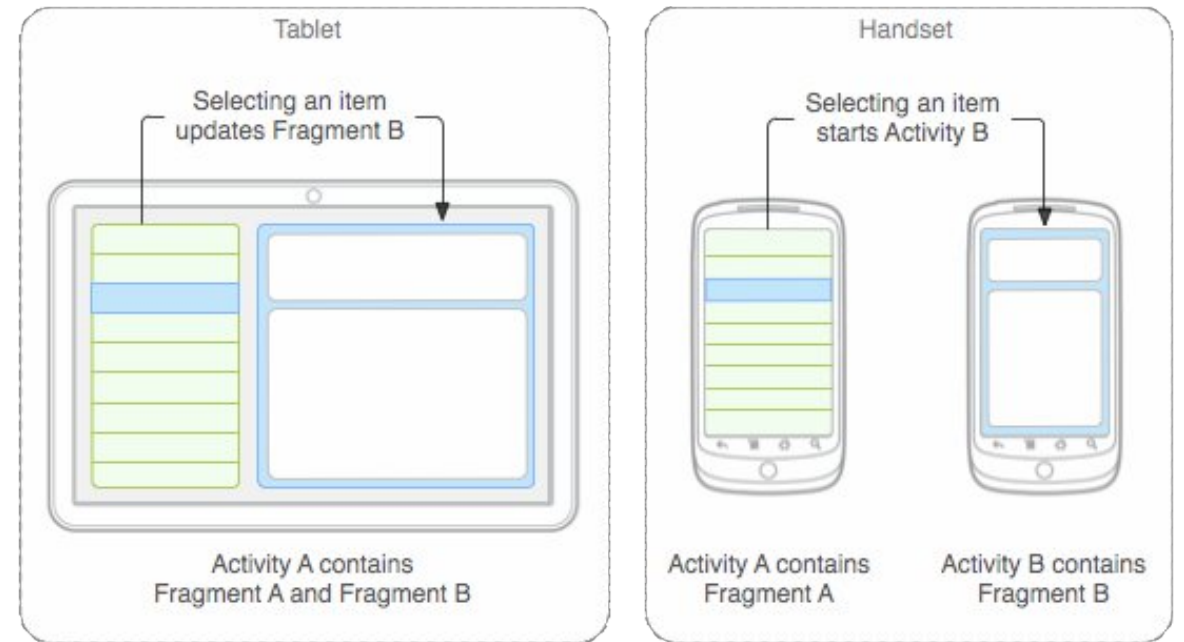
Características

Foram introduzidos com o Android 3.0, de modo a fornecer uma interface mais dinâmica e flexível;

Existe uma biblioteca de suporte (android-support-v4.jar) que permite utilizar Fragments em versões anteriores à 3.0;

Por exemplo, numa aplicação de notícias em Tablet:

- À esquerda, um Fragment para mostrar uma lista com várias notícias
- À direita, um Fragment que apresenta o conteúdo da notícia selecionada



Fragments

Características (2)

Recebem os seus **próprios** inputs (Intents);

Possuem o seu **próprio** ciclo de vida e o seu **próprio** layout;

Podem ser **adicionados** ou **removidos** de uma Activity em tempo real;

Criados para serem **reutilizados**;

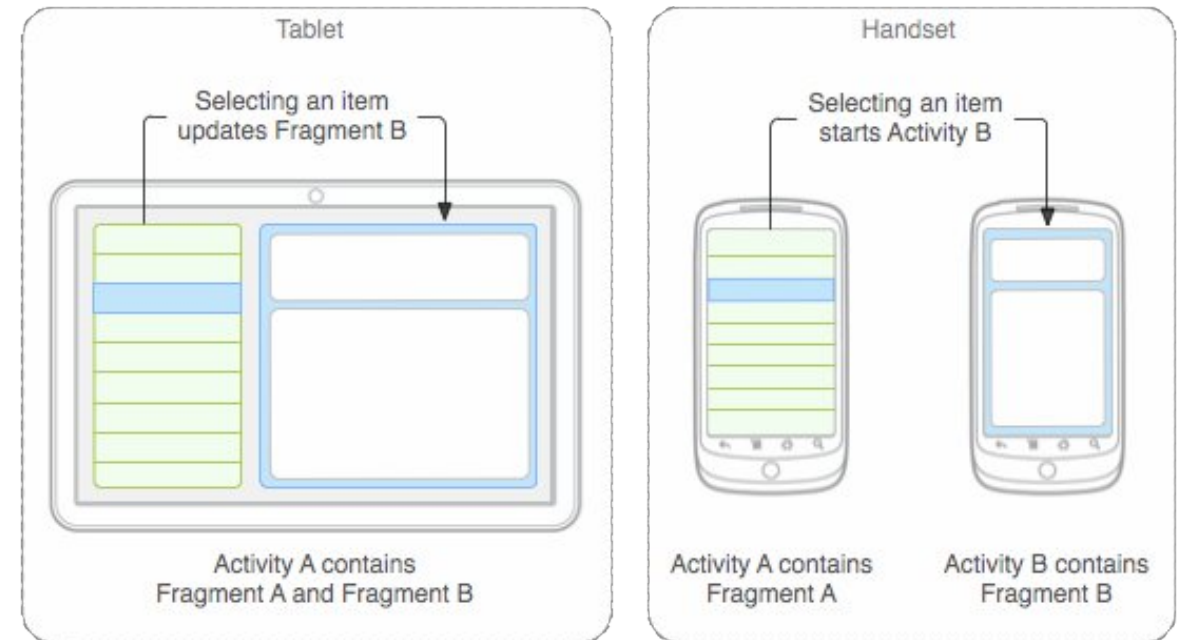
Têm de ser embebidos numa Activity e o seu ciclo de vida é **diretamente afetado** pelo ciclo de vida da Activity:

- Quando a Activity passa pelo `onPause()` **todos** os Fragments no interior da Activity também o fazem.

Fragments

Importância de utilização

- Reutilização de views e componentes lógicos;
- Suporte a tablets;
- Orientação de ecrã;



Fragments

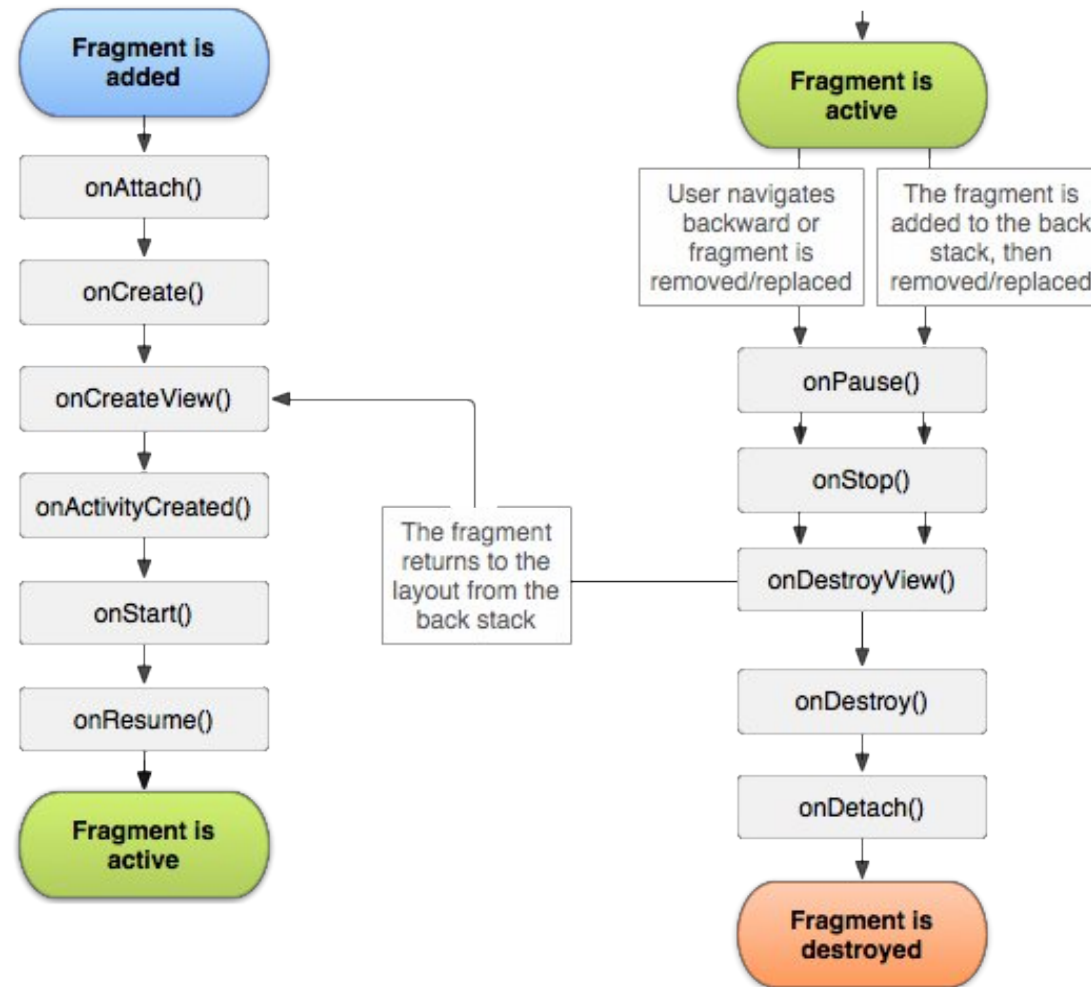
Diferenças de uma Activity

Não representam um contexto, pelo que convém obter sempre o contexto da Activity e guardá-lo numa variável;

Atribuição do layout não é efetuada através do método `setContentView`, mas sim utilizando o `LayoutInflater` do método `onCreateView`;

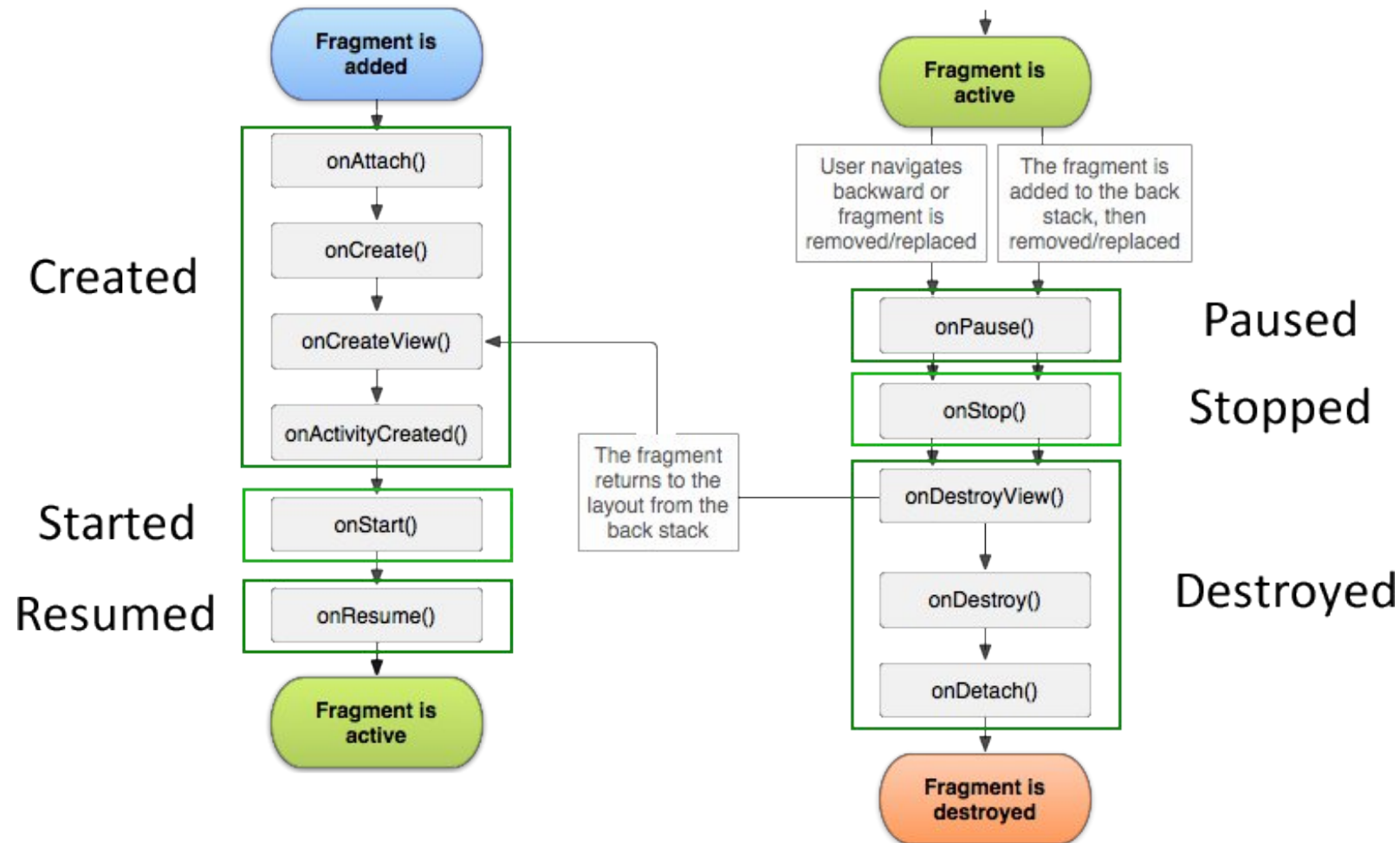
Ciclo de vida ligeiramente diferente embora **dependente** do ciclo de vida da Activity em que se insere.

Ciclo de Vida



Cada Fragment tem um **ciclo de vida próprio...**

Ciclo de Vida



Cada Fragment tem um **ciclo de vida próprio...** que **interage** com o da Activity em que se insere.

Fragments

Ciclo de Vida -> Novos Métodos

- `onAttach` – Invocado quando o Fragment é associado a uma Activity
- `onCreateView` – É invocado quando a interface gráfica do Fragment vai ser desenhada. Este método deve devolver uma View, ou caso seja um Fragment sem interface gráfica, deve devolver `null`;
- `onActivityCreated` – Invocado imediatamente após a execução do método `onCreate` da Activity;
- `onDestroyView` – O inverso do `onCreateView`, ou seja, é invocado quando os componentes gráficos do Fragment vão ser removidos da interface gráfica;
- `onDetach` – O inverso do `onAttach`, é invocado quando o Fragment vai ser desassociado da Activity.

Fragments

Criação

A superclasse deve ser `Fragment` ou uma das suas subclasses:

- `DialogFragment` – cria uma dialog flutuante
- `ListFragment` – similar à `ListView`
- `PreferenceFragment` – similar à `PreferenceActivity` (só existe a partir do Android 3.0)
- `WebViewFragment` – mostra uma `WebView`, para visualizar websites

A estrutura de código é idêntica à de uma `Activity`;

Pode-se inclusive facilmente **converter** uma `Activity` **existente** num `Fragment`.

!! Não precisa de ser declarado no `AndroidManifest.xml` !!

Fragments

fragment_example.xml

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

    <TextView
        android:id="@+id/txt_example"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/example" />

    <Button
        android:id="@+id/btn_example"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/btn_example" />

</LinearLayout>
```

Fragments

FragmentExample.java

```
public class FragmentExample extends Fragment {

    private Context mContext;
    private TextView txtViewExample;
    private Button btnExample;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mContext = getActivity();
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View mContentView = inflater.inflate(R.layout.fragment_example, container, false);
        txtViewExample = (TextView) mContentView.findViewById(R.id.txt_example);
        btnExample = (Button) mContentView.findViewById(R.id.btn_example);

        return mContentView;
    }
}
```

Fragments

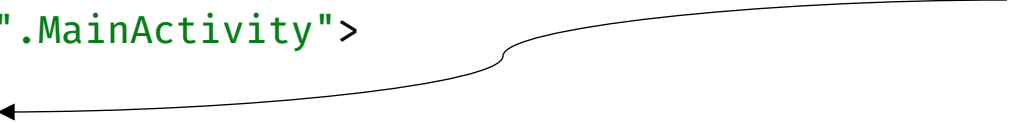
activity_main.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <fragment
        android:id="@+id/fragment_test"
        android:name="pt.ipp.estg.t_fragment.FragmentExample"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

Inserção de forma estática



Fragments

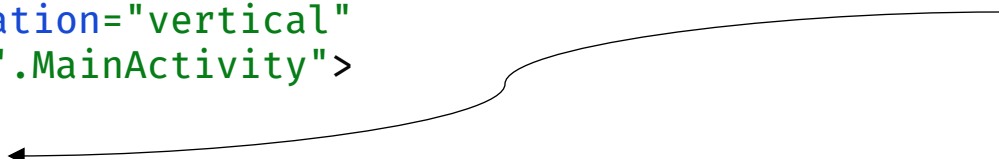
activity_main.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <FrameLayout
        android:id="@+id/fragment_test"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

Inserção de forma dinâmica



Fragments

Atributos Importantes

android:name – é aqui que é especificado qual o Fragment que irá ser colocado no layout da Activity. Corresponde ao identificador da classe Java do Fragment (package + nome)

[Apenas se aplica na inserção estática de Fragments]

android:id – necessário para termos uma referência no código da Activity para o Fragment no Layout em XML;

android:tag – identificador único (string) para representar o Fragment;

Fragments

MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        FragmentExample fragmentExample = new FragmentExample();  
  
        FragmentTransaction fragmentTransaction = getSupportFragmentManager().beginTransaction();  
        fragmentTransaction.replace(R.id.fragment_container, fragmentExample);  
        fragmentTransaction.addToBackStack(null);  
        fragmentTransaction.commit();  
    }  
}
```

Fragments

Gestão

Os Fragment são **adicionados, removidos e substituídos** numa Activity utilizando a classe `FragmentManager`.

Esta é inicializada da seguinte forma

```
FragmentManager fragmentManager = getSupportFragmentManager();  
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
```

Não esquecer de fazer commit no final da transação

```
fragmentTransaction.commit();
```

Fragments

Gestão (2)

Adicionar Fragment

```
fragmentTransaction.add(ID_VIEW_FRAGMENT, FRAGMENT);
```

Substituir Fragment

```
fragmentTransaction.replace(ID_VIEW_FRAGMENT, FRAGMENT);
```

Remover Fragment

```
fragmentTransaction.remove(FRAGMENT);
```

Utilizando o método `fragmentTransaction.addToBackStack(String)`, antes de fazer `fragmentTransaction.commit()`, podemos adicionar o Fragment à stack de Activities.

Para retirar um Fragment da stack podemos invocar o método `popBackStack()`.

Fragments

Encontrar Fragment por ID

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        FragmentExample fragmentExample = new FragmentExample();  
  
        FragmentManager fragmentManager = getSupportFragmentManager();  
  
        fragmentManager.findFragmentById(R.id.fragment_container);  
    }  
}
```

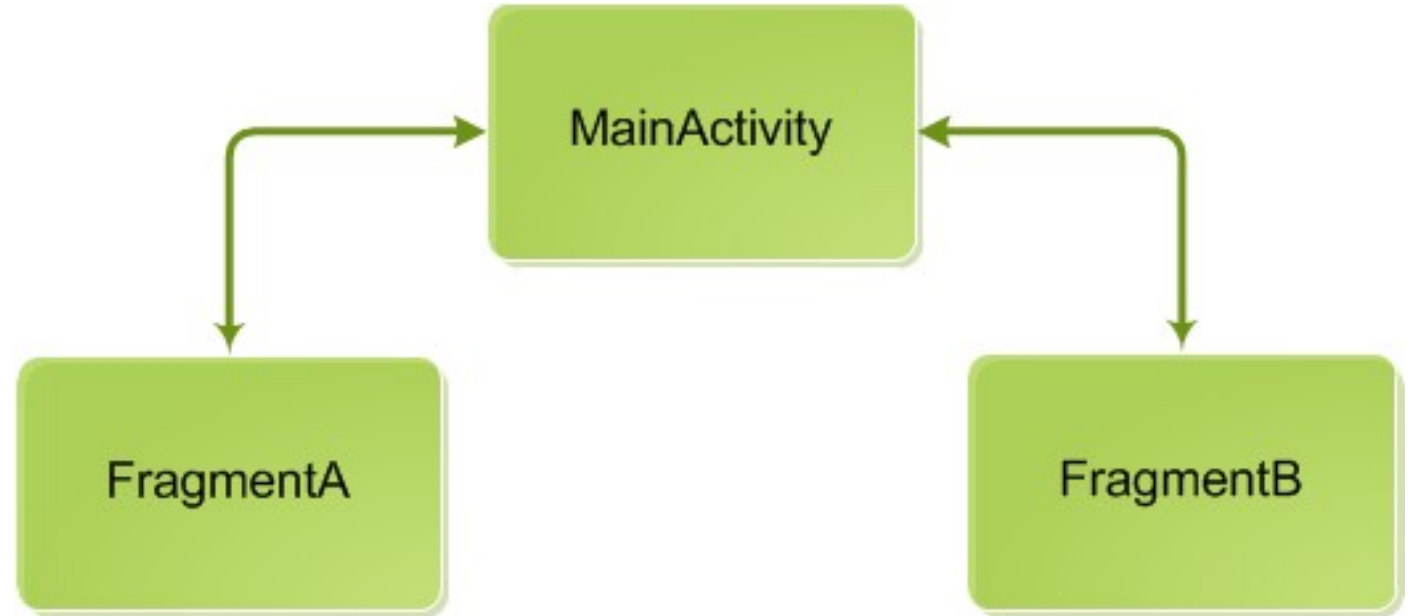
Fragments

Encontrar Fragment por TAG

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        FragmentExample fragmentExample = new FragmentExample();  
  
        FragmentManager fragmentManager = getSupportFragmentManager();  
  
        fragmentManager.findFragmentByTag("FRAGMENT_TAG");  
    }  
}
```

Comunicação entre Fragments

Fragments **não conseguem comunicar entre si**, para o fazerem têm de comunicar **através da Activity**, que depois comunica com o outro Fragment.



Comunicação entre Fragments

App Lista de Países do Mundo

Vamos imaginar uma aplicação que lista os detalhes de países de todo o mundo. Podemos ter os seguintes componentes:

- MainActivity: que incorpora dois Fragments;
- CountriesListFragment: FragmentList com a lista de países;
- DetailsFragment: Fragment que mostra os detalhes do país escolhido;
- OnCountrySelectedListener interface: para haver comunicação entre Fragment e Activity.

Comunicação entre Fragments

interface OnCountrySelectedListener

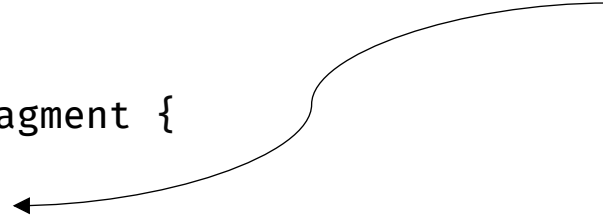
```
static interface OnCountrySelectedListener {  
    //Método utilizado para notificar a instância que está a "escuta"  
    public void onCountrySelected(int position);  
}
```

Comunicação entre Fragments

CountryListFragment.java

A MainActivity.java irá registrar-se no CountryListFragment de forma a executar eventos de seleção de um país

```
public class CountriesListFragment extends ListFragment {  
  
    static interface OnCountrySelectedListener{  
        public void OnCountrySelected(long id);  
    }  
  
    private OnCountrySelectedListener listener;
```



Comunicação entre Fragments

CountryListFragment.java

```
@Nullable
@Override
public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
                        @Nullable Bundle savedInstanceState) {
    String[] countryNames = new String[Country.countries.length];
    for(int i = 0; i < countryNames.length; i++){
        countryNames[i] = Country.countries[i].getName();
    }

    ArrayAdapter<String> adapter = new ArrayAdapter<String>(
        inflater.getContext(),
        android.R.layout.simple_list_item_1,
        countryNames
    );
    setListAdapter(adapter);

    return super.onCreateView(inflater, container, savedInstanceState);
}
```

Comunicação entre Fragments

CountryListFragment.java

No método onAttach() verificamos se a MainActivity está a implementar a interface

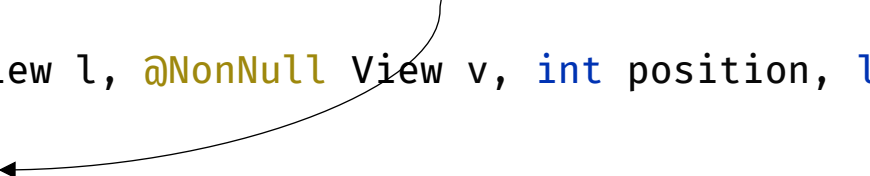
```
@Override
public void onAttach(@NonNull Context context) {
    super.onAttach(context);
    try{
        this.listener = (OnCountrySelectedListener) context;
    }catch (ClassCastException e){
        throw new ClassCastException(getActivity().toString() + "must implement OnCountrySelectedListener");
    }
}
```

Comunicação entre Fragments

CountryListFragment.java

O FragmentList já possui um listener de eventos (click). Sempre que houver uma interação com um item, invocamos o método OnCountrySelected() presente na MainActivity.

```
@Override
public void onListItemClick(@NonNull ListView l, @NonNull View v, int position, long id) {
    if(listener != null){
        listener.OnCountrySelected(id);
    }
}
```



Comunicação entre Fragments

DetailsFragment.java

```
private long countryId;
```

```
public void setCountryId(long id){  
    this.countryId = id;  
}
```

Registamos uma variável que representa o id do objeto que queremos apresentar, e um método para alterar esse valor.

Comunicação entre Fragments

MainActivity.java

Implementamos a interface

```
public class MainActivity extends AppCompatActivity implements CountriesListFragment.OnCountrySelectedListener {
```

Implementamos o método da interface, que
será responsável por toda a comunicação

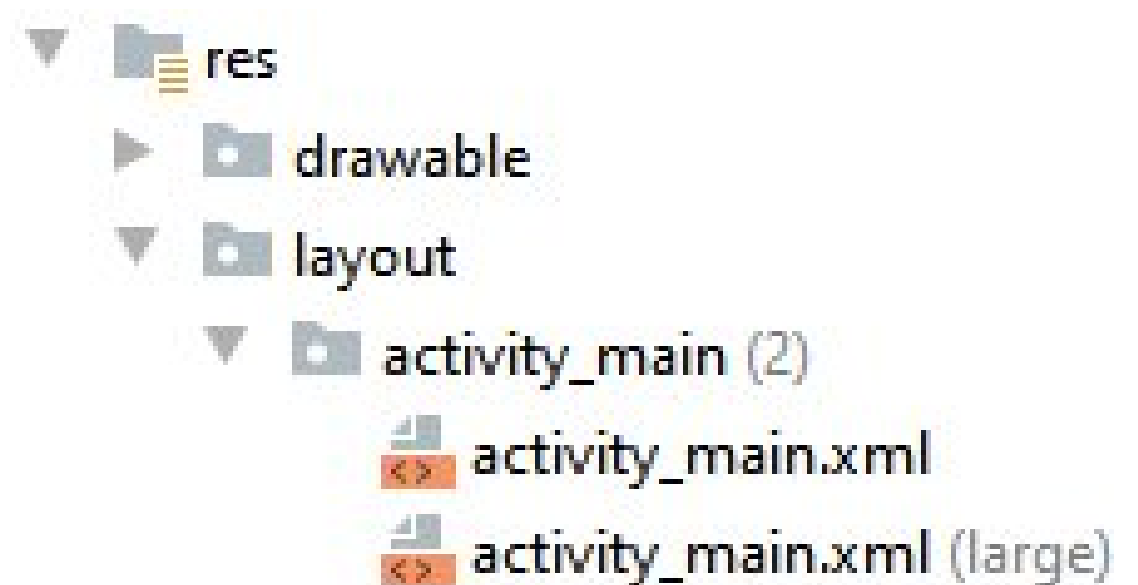
@Override

```
public void OnCountrySelected(long id) {  
    DetailsFragment detailsFragment = new DetailsFragment();  
    detailsFragment.setCountryId(id);  
  
    fragmentTransaction = fragmentManager.beginTransaction();  
    fragmentTransaction.replace(R.id.fragment_container, detailsFragment);  
    fragmentTransaction.addToBackStack(null);  
    fragmentTransaction.commit();  
}
```

Tablet e Smartphone

Utilização automática de diferentes layouts conforme o dispositivo seja uma Tablet ou um Smartphone

- Diferentes ficheiros de layout
- Com o mesmo nome
- Mas com configurações distintas

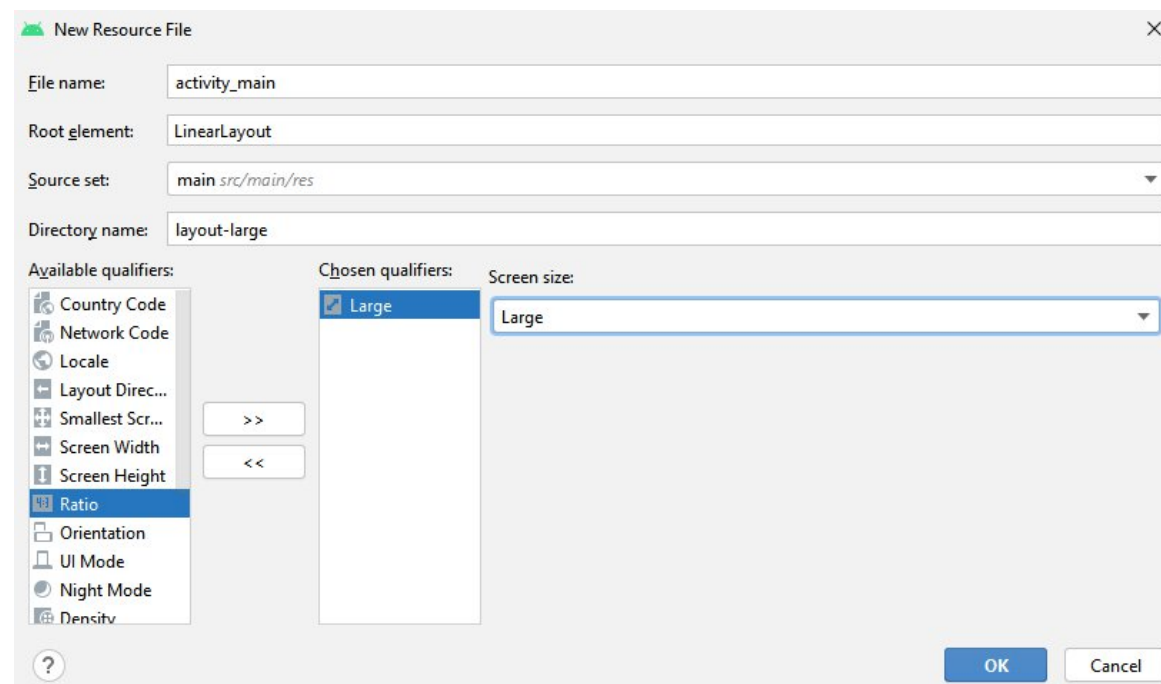
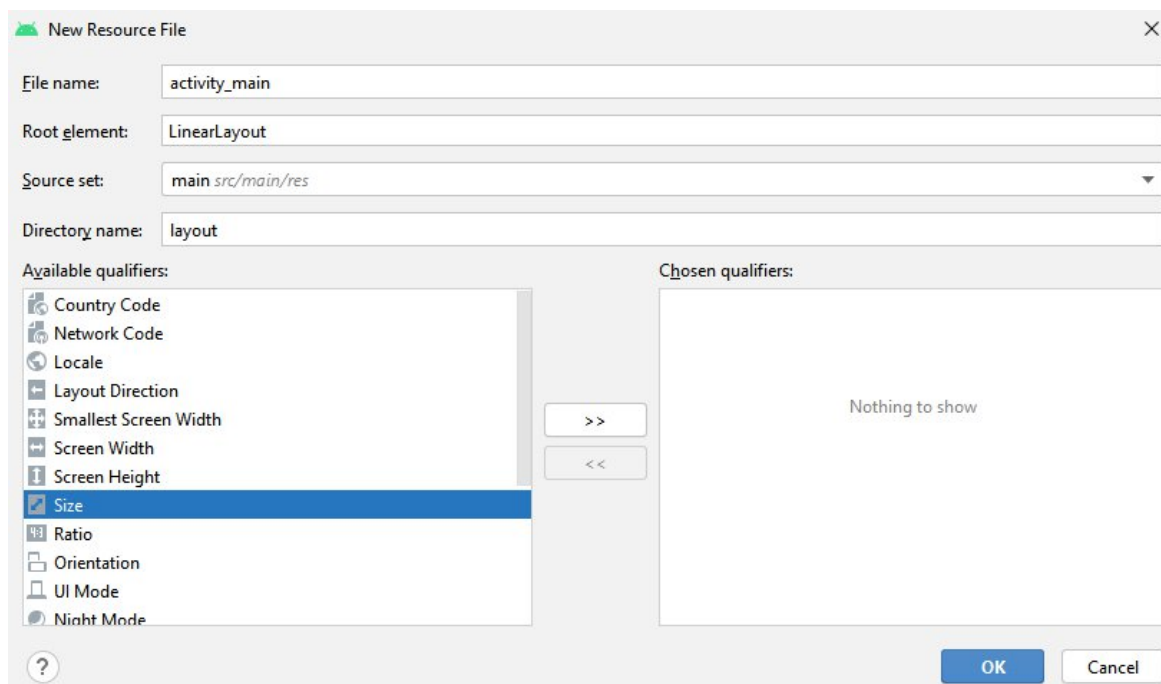


Tablet e Smartphone

Criação do Layout

Ao criar um **novo layout** podemos especificar qual o **tamanho** a que o mesmo se adequa.

Na configuração do layout escolhemos o qualifier **Size** e depois o tamanho **Large**



Tablet e Smartphone

activity_main.xml [smartphone]

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

Tablet e Smartphone

activity_main.xml (large) [tablet]

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

    <fragment
        android:name="pt.ipp.estg.t_fragment.CountriesListFragment"
        android:id="@+id/fragment_countries"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1" />

    <fragment
        android:name="pt.ipp.estg.t_fragment.DetailsFragment"
        android:id="@+id/fragment_details"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="2" />

</LinearLayout>
```

Tablet e Smartphone

MainActivity.java [verificação de layout a utilizar]

```
private FragmentManager fragmentManager;  
private FragmentTransaction fragmentTransaction;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    if(findViewById(R.id.fragment_container) != null){  
        if(savedInstanceState != null){  
            return;  
        }  
        CountriesListFragment countriesListFragment = new CountriesListFragment();  
        fragmentManager = getSupportFragmentManager();  
        fragmentTransaction = fragmentManager.beginTransaction();  
        fragmentTransaction.add(R.id.fragment_container, countriesListFragment);  
        fragmentTransaction.addToBackStack(null);  
        fragmentTransaction.commit();  
    }  
}
```

Se tivermos a utilizar um dispositivo Tablet, esta View não será inicializada (ou seja, será null)

Tablet e Smartphone

MainActivity.java [ações variam dependendo do layout]

```
@Override
public void onCountrySelected(long id) {
    DetailsFragment detailsFragment =
        (DetailsFragment) getSupportFragmentManager().findFragmentById(R.id.fragment_details);

    if(detailsFragment != null){
        detailsFragment.updateCountry(id);
    }else{
        detailsFragment = new DetailsFragment();
        detailsFragment.setCountryId(id)

        fragmentTransaction = fragmentManager.beginTransaction();
        fragmentTransaction.replace(R.id.fragment_container, detailsFragment);
        fragmentTransaction.addToBackStack(null);
        fragmentTransaction.commit();
    }
}
```

Verificação se estamos a utilizar o layout de tablet. Em caso afirmativo este valor de id da view será diferente de null

Tablet e Smartphone

App final [smartphone]

T_fragment	
Country :	Portugal
Continent :	Europe
Country :	United Kingdom
Continent :	Europe
Country :	Mexico
Continent :	America
Country :	Argentina
Continent :	America
Country :	Brazil
Continent :	America
Country :	Egypt
Continent :	Africa
Country :	Sidney
Continent :	Australia
Country :	Greece
Continent :	Europe

T_fragment	
Country :	Portugal
Continent :	Europe
<p>Portugal is a southern European country on the Iberian Peninsula, bordering Spain. Its location on the Atlantic Ocean has influenced many aspects of its culture: salt cod and grilled sardines are national dishes, the Algarve's beaches are a major destination and much of the nation's architecture dates to the 1500s–1800s, when Portugal had a powerful maritime empire.</p>	

Tablet e Smartphone

App final [tablet]

T_fragment			
Country :	Portugal	Country :	Portugal
Continent :	Europe	Continent :	Europe
Country :	United Kingdom	<p>Portugal is a southern European country on the Iberian Peninsula, bordering Spain. Its location on the Atlantic Ocean has influenced many aspects of its culture: salt cod and grilled sardines are national dishes, the Algarve's beaches are a major destination and much of the nation's architecture dates to the 1500s–1800s, when Portugal had a powerful maritime empire.</p>	
Continent :	Europe		
Country :	Mexico		
Continent :	America		
Country :	Argentina		
Continent :	America		
Country :	Brazil		
Continent :	America		
Country :	Egypt		
Continent :	Africa		
Country :	Sidney		
Continent :	Australia		
Country :	Greece		
Continent :	Europe		

Leitura Adicional

Fragments

<https://developer.android.com/guide/components/fragments>

Comunicação entre Fragments

<https://developer.android.com/training/basics/fragments/communicating>

UI para Tablets e Smartphones

<https://developer.android.com/training/basics/fragments/fragment-ui>

Programação Para Dispositivos Móveis I

FRAGMENTS

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