



# Análise e Desenvolvimento de Sistemas

Mobile Development and IOT - Android

Fragments

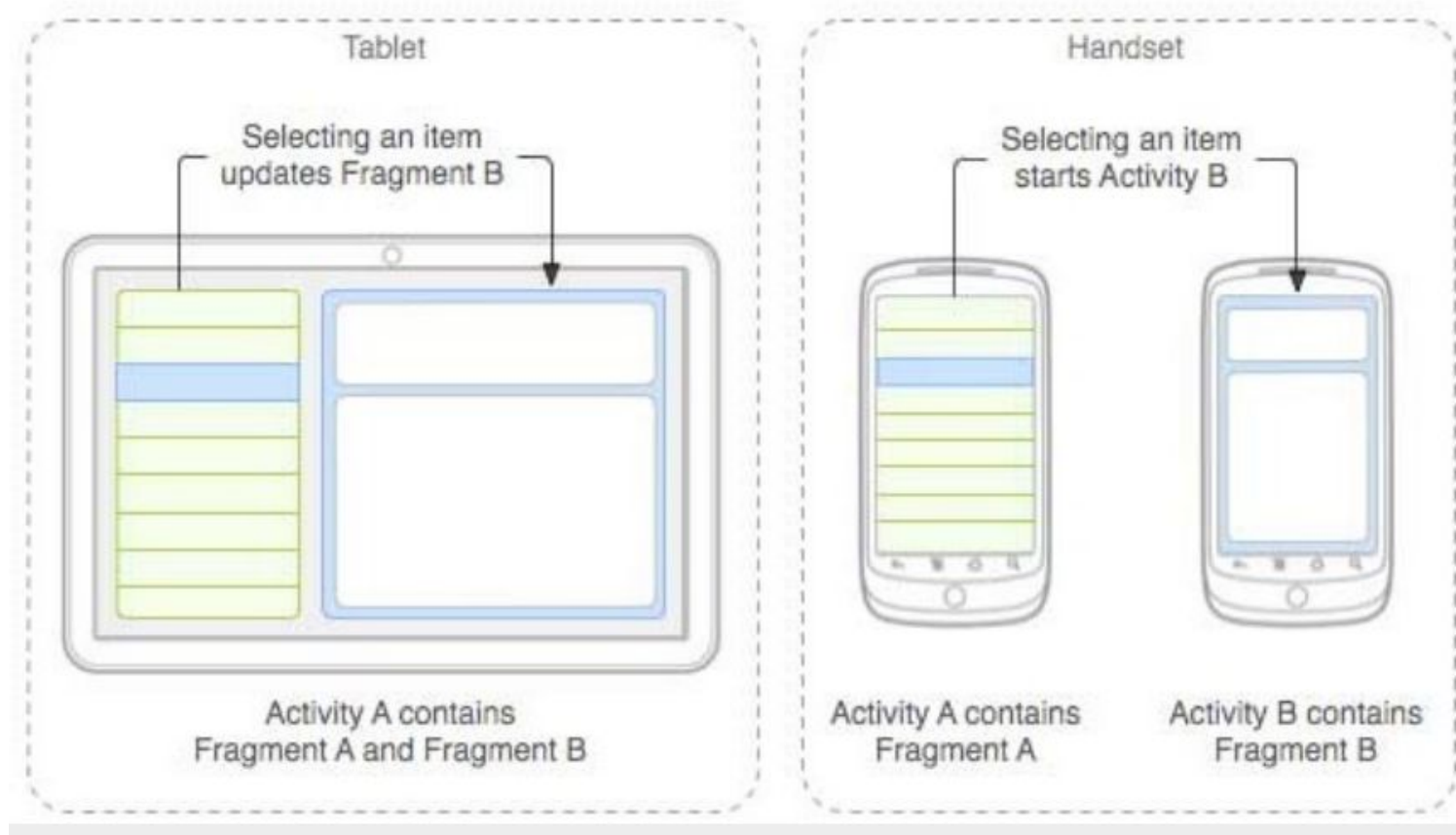


Prof. Douglas Cabral <[douglas.cabral@fiap.com.br](mailto:douglas.cabral@fiap.com.br)>  
<https://www.linkedin.com/in/douglascabral/>

Na plataforma Android, o conceito de **Fragments** nos permite **modularizar** nossas Activities e o **reaproveitamento** de componentes e lógicas em diversas outras Activities.

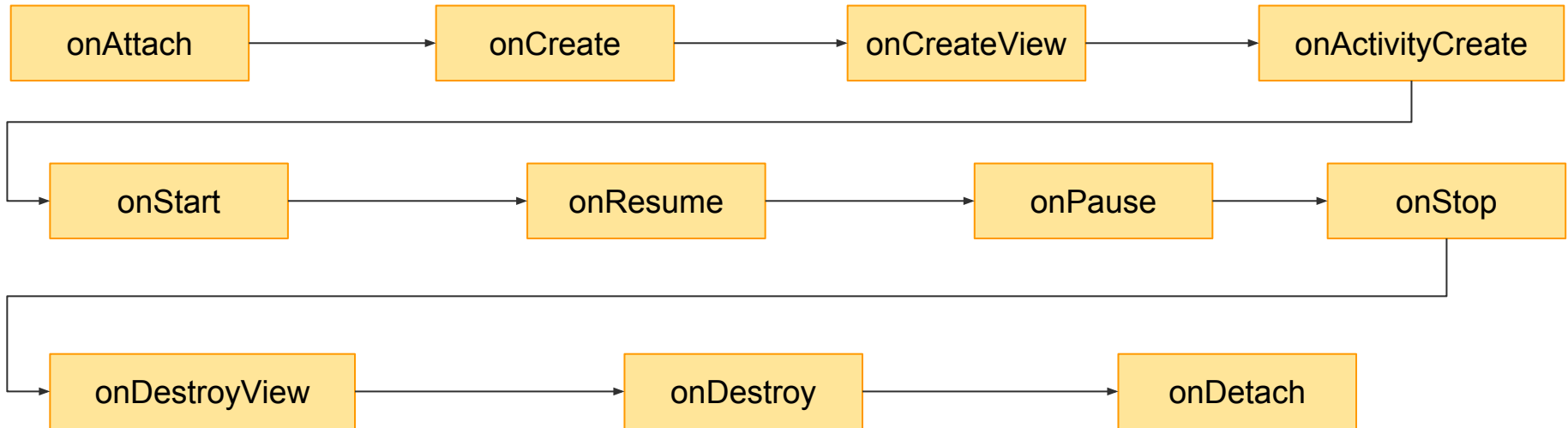
Este conceito chegou a partir do **Android Honeycomb (Android 3.0 - API Level 11)**, como uma forma de desenvolver layouts adaptados para tablets (que possuem tela maior) e celulares (telas menores), utilizando o conceito layout master/detail.

Nos tablets poderiam ser utilizados 02 fragments ou mais, enquanto no celular era exibido apenas um fragment para seleção da opção e o conteúdo aparecia conforme o clique na primeira fragment.



- ❑ É um tipo de sub-activity
- ❑ Layout, comportamento e ciclo de vidas próprios
- ❑ É possível adicionar/remover fragments durante execução
- ❑ Um fragmento pode ser reutilizado em diferentes activities
- ❑ Uma activity pode possuir diversos fragmentos
- ❑ Um fragmento tem dependência com a Activity master do qual foi chamado

Os **fragments** são executados dentro do contexto da Activity, porém, **possuem um ciclo de vida próprio**, bem **parecido com o ciclo de vida da Activity** e podem possuir uma interface (layout) própria.



## Fragments - Funcionamento - Ciclo de vida (Diferenças)



**onCreateView** - Onde é inflado/criado a layout do fragment

**onAttach** - Evento disparado ao ser inserido em uma Activity

**onDetach** - Evento disparado ao ser removido de uma Activity - Última coisa a acontecer após a destruição do Fragment.

**onDestroyView** - Local para limpeza de recursos em memória antes do Fragment ser destruído

++

45697096

# Fragment estático



45697096

45697096



Para criar uma fragment, basta criarmos uma classe que estende a classe **Fragment**.

Ao fazer o import da classe, dê preferência ao pacote de support, que faz com que a Fragment possua mais compatibilidade com as diferentes versões do sistema operacional.

```
import android.support.v4.app.Fragment;  
  
public class MeuFragment extends Fragment {  
}
```



## Fragments - Criação

Vamos criar uma layout para este fragment em **res/layout/fragment\_meu.xml** e inflá-lo na classe da Fragment.

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

</LinearLayout>
```

```
public class MeuFragment extends Fragment {
    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_meu, container, attachToRoot: false);
    }
}
```

Na **layout** da nossa **Main Activity** vamos colocar uma **tag fragment** com um **atributo class**, no qual, descreve o caminho da classe referente ao nosso fragment recém criado.

**Obs:** Altere para o seu endereço de pacote ou caminho caso necessário no atributo **class**.

```
<fragment
    android:id="@+id/frgMain"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    class="br.com.fiap.fragments.MeuFragment"
/>
```



## Fragments - Resultado final

Insira algum conteúdo na layout de sua Fragment e execute a aplicação para ver o resultado final.



+ +

45697096

# FragmentManager



Podemos **trocar dinamicamente uma fragment** utilizando a classe **FragmentManager**.

O **FragmentManager** pode ser acessado na Activity por **getSupportFragmentManager** (quando utilizado o pacote de suporte) ou **getFragmentManager**.

# FragmentManager

Vamos criar nossa MainActivity com  
**02 botões e um FrameLayout**,  
que será o container para a troca  
de nossos **fragments**.

Veja que o id do FrameLayout é  
**fragmentContainer**, mas nada impede  
de ser outro nome.

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

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Exibir fragment 1"
        android:onClick="exibirFragment1"/>

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Exibir fragment 2"
        android:onClick="exibirFragment2"/>

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

</LinearLayout>
```

Crie outra **fragment** chamada OutroFragment com seu respectivo layout e nos eventos de cada botão vamos inserir:

```
public void exibirFragment1(View view) {  
    getSupportFragmentManager()  
        .beginTransaction()  
        .replace(R.id.fragmentContainer, new MeuFragment())  
        .addToBackStack(null)  
        .commit();  
}
```

O primeiro parâmetro do método replace é o ID do elemento (FrameLayout) que irá receber o fragment.

```
public void exibirFragment2(View view) {  
    getSupportFragmentManager()  
        .beginTransaction()  
        .replace(R.id.fragmentContainer, new OutroFragment())  
        .addToBackStack(null)  
        .commit();  
}
```

O segundo parâmetro é a instância do Fragment.

Além do método **replace** visto anteriormente, no **FragmentManager** temos também os métodos **add** e **remove**.

```
getSupportFragmentManager()  
    .beginTransaction()  
    .add(R.id.fragmentContainer, new MeuFragment())  
    .addToBackStack(null)  
    .commit();
```

Exemplo de add

---

```
Fragment fg = getSupportFragmentManager().findFragmentById(R.id.fragmentContainer);
```

```
getSupportFragmentManager()  
    .beginTransaction()  
    .remove(fg)  
    .addToBackStack(null)  
    .commit();
```

Exemplo de remove



++

45697096

# Exercício



Crie um aplicativo de carros, onde ao clicar em cada botão ou imagem de carro, seja exibido suas características dentro de Fragments.

+

# Dúvidas?





Copyright © 2018 Prof. Douglas Cabral <[douglas.cabral@fiap.com.br](mailto:douglas.cabral@fiap.com.br)> <https://www.linkedin.com/in/douglascabral/>

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).