

Intents



Prof. Windson Viana
Prof. Fernando Trinta

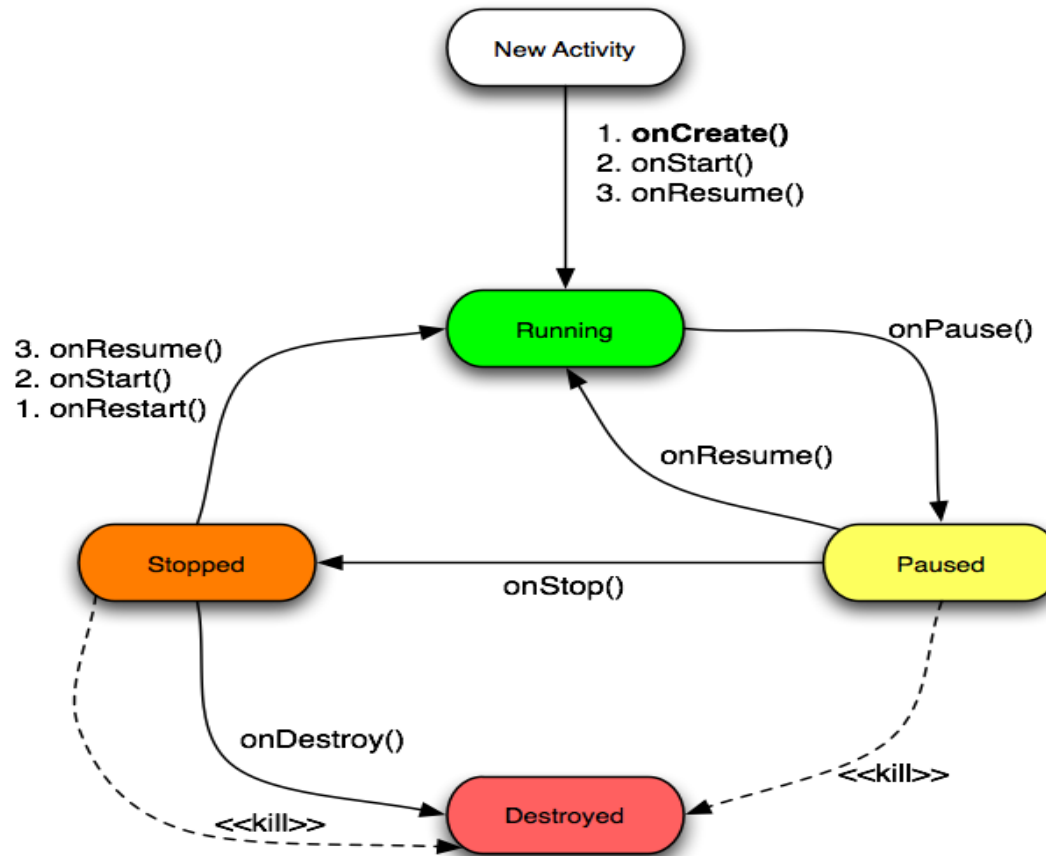


Disciplina de Introdução à Computação Móvel e Ubíqua – 2014.1
Curso de Sistemas e Mídias Digitais

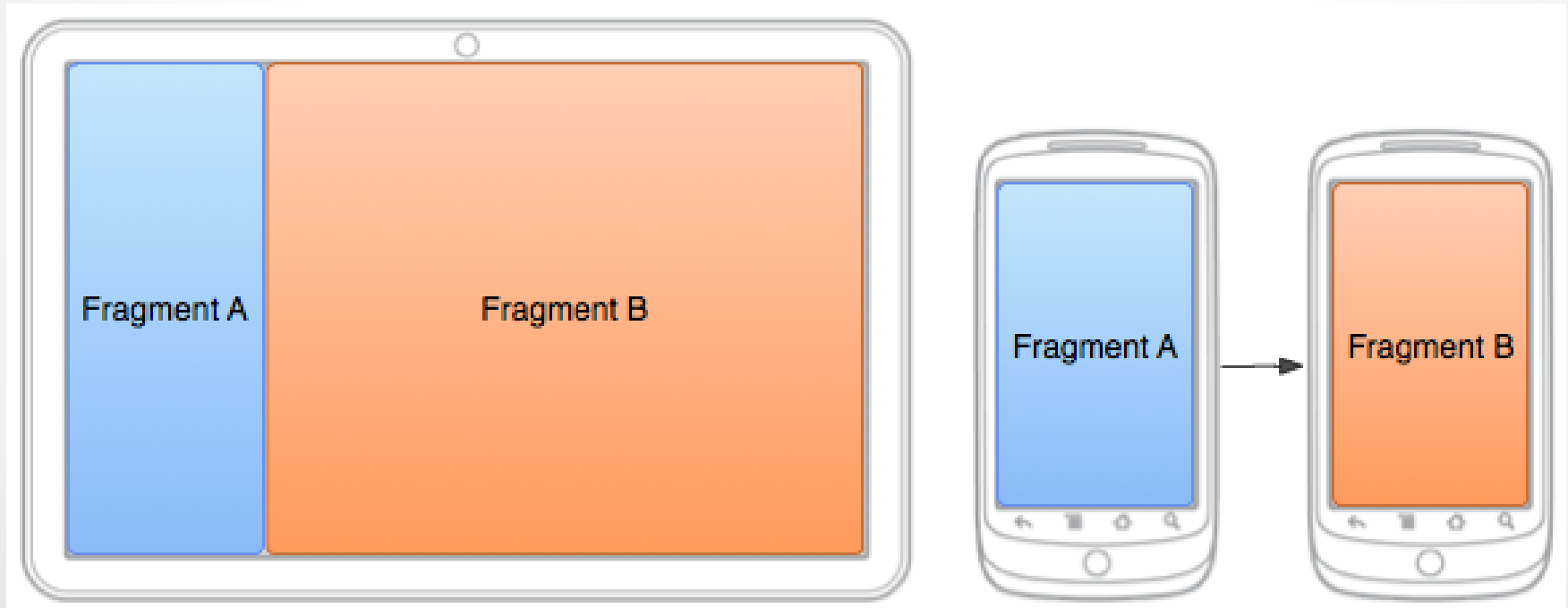
Disciplina Redes Móveis Sem Fio – 2014.1
Mestrado e Doutorado em Ciência da Computação

Ciclo de Vida de uma Activity

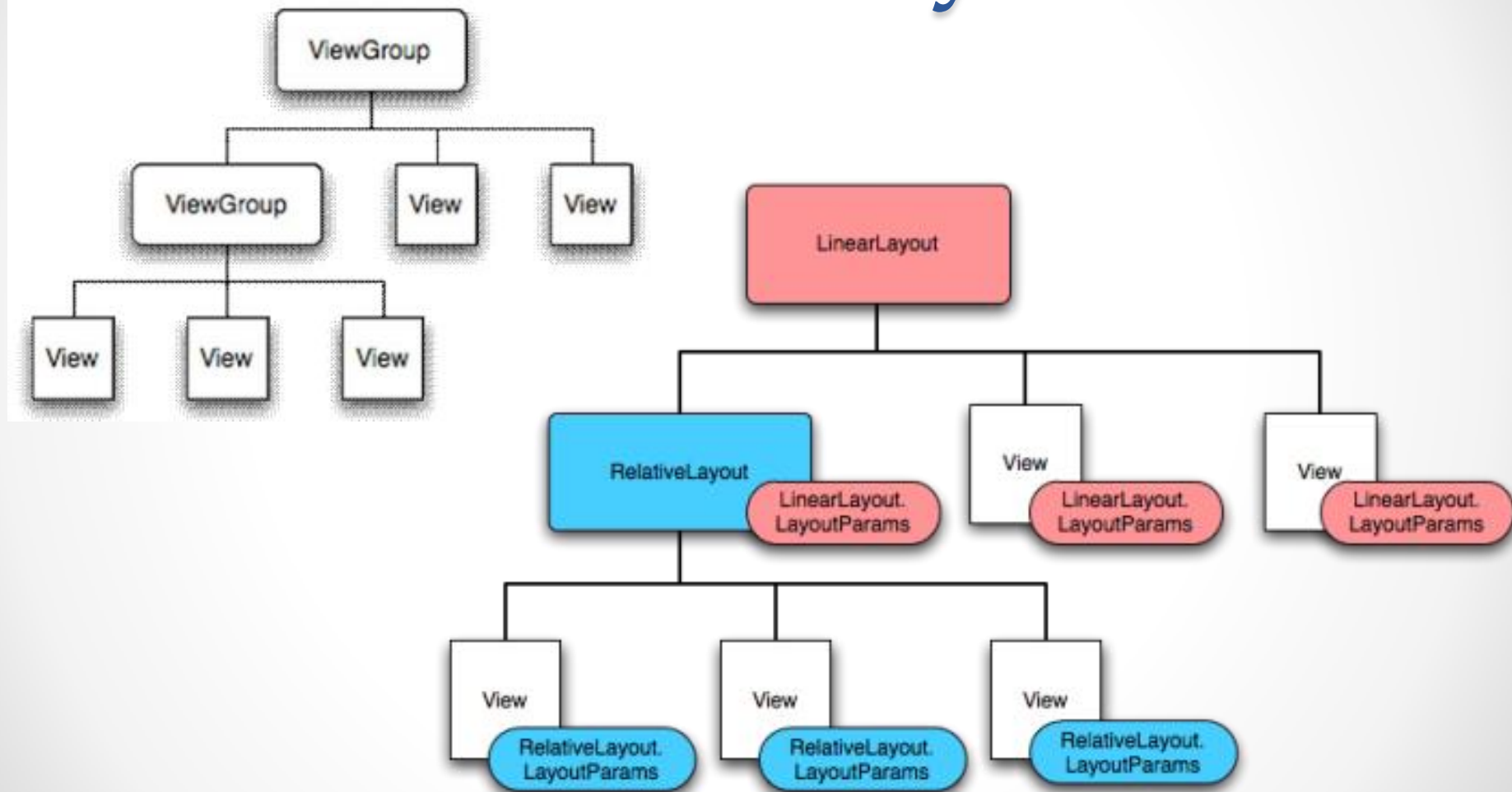
Activity Lifecycle



Fragments



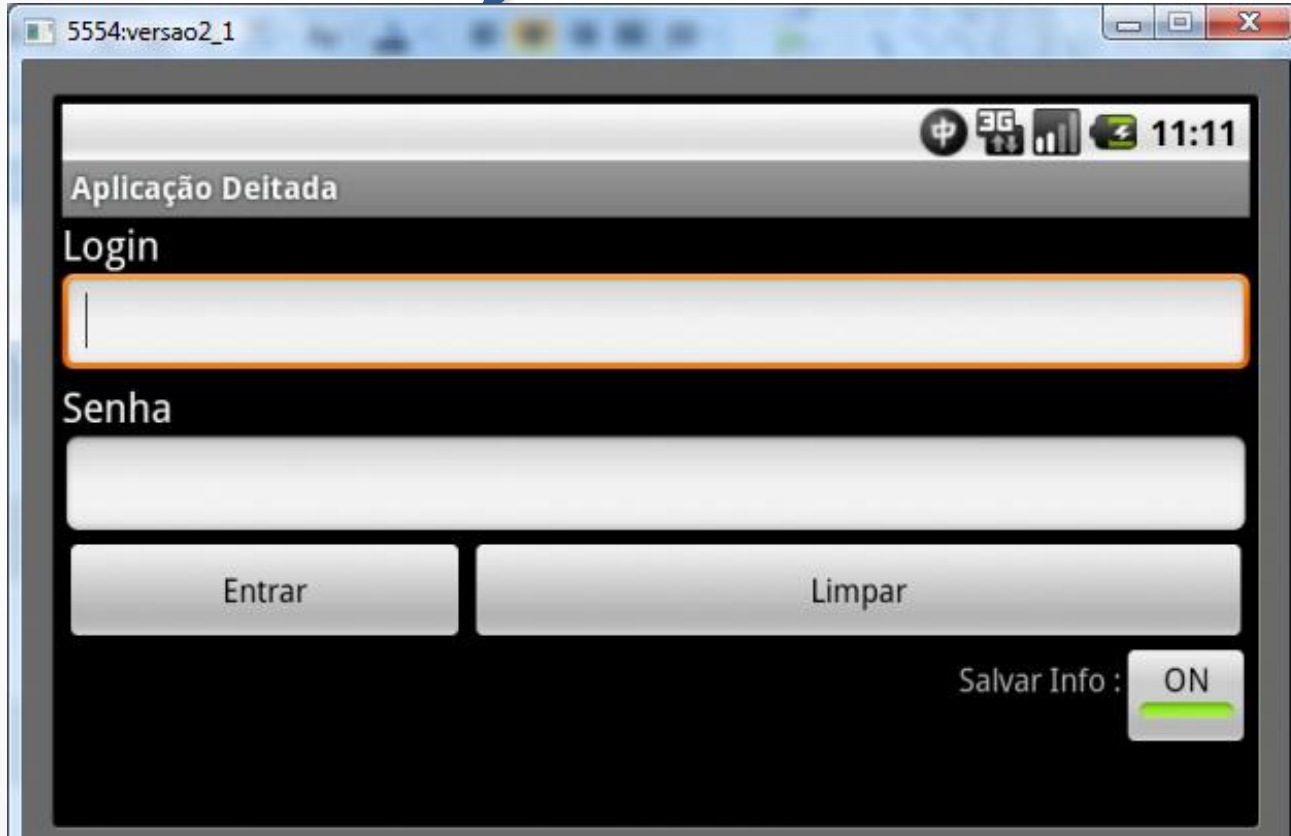
Tipos de Layout



Tipos de Layout

- LinearLayout
 - Cada componente em uma linha (vertical) ou
 - Cada componente em uma coluna (horizontal)
 - Parâmetros
 - `android:layout_width=fill_parent`
 - `android:layout_height="wrap_content"`
 - `android:orientation="vertical"`
 - `android:layout_weight="0.3"`
 - `android:gravity="right"`

LinearLayout - Exercício



Monte essa interface usando 3 LinearLayout

Como salvar as Informações?

- Classe SharedPreferences
 - Mecanismo simples para salvar um conjunto de pares atributo-valor
 - Compartilhar dados entre atividades
 - Exemplo, salvar login e senha

Como salvar as Informações?

```
private void salvarLoginESenha() {  
    int mode= Activity.MODE_PRIVATE; //Quem pode acessar?  
    SharedPreferences prefLogin=  
        getSharedPreferences("prefLogin", mode);  
    SharedPreferences.Editor editor= prefLogin.edit();  
    editor.putString("login", "windson");  
    editor.commit();  
}  
  
private String lerLogin() {  
    int mode= Activity.MODE_PRIVATE; //Quem pode acessar?  
    SharedPreferences prefLogin=  
        getSharedPreferences("prefLogin", mode);  
    return prefLogin.getString("login", "");  
}
```


Exercício

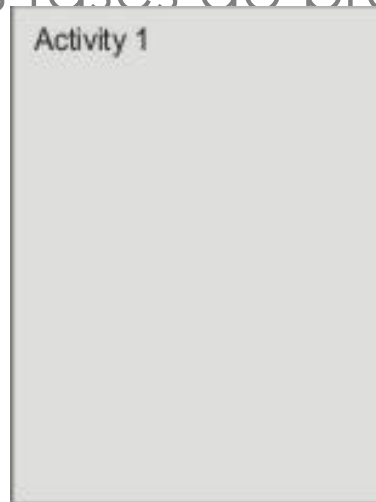
- Salve o login e a senha digitados se o ToggleButton estiver selecionado!

ImageButton

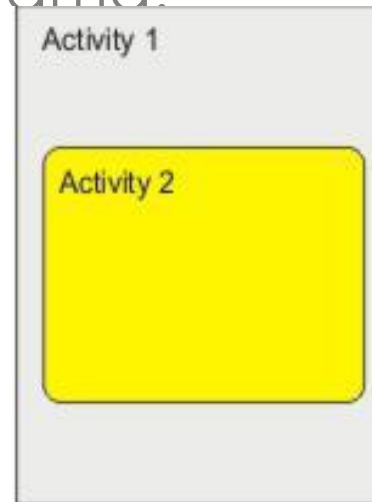
- Funciona como um botão tradicional com uma imagem.
- `<ImageButton`
- `android:id="@+id/imageButton1"`
- `android:layout_width="wrap_content"`
- `android:layout_height="wrap_content"`
- **`android:background="#00FFFFFF"`**
- `android:src="@drawable/logoAri" />`

Activity

- Uma Activity (ou Atividade) representa uma interface gráfica com o usuário. Aplicações podem definir uma ou mais atividades para manipular diferentes fases do programa.



Activity em foco



Activity em modo 'Pause'

Activity 2 entra na frente da Activity 1. Porém ela ainda é visível. Apenas `onPause()` é chamada.



Activity em modo 'Stop'

Activity 3 entra na frente da Activity 1, e ela não é mais visível. `onStop()` é chamada.

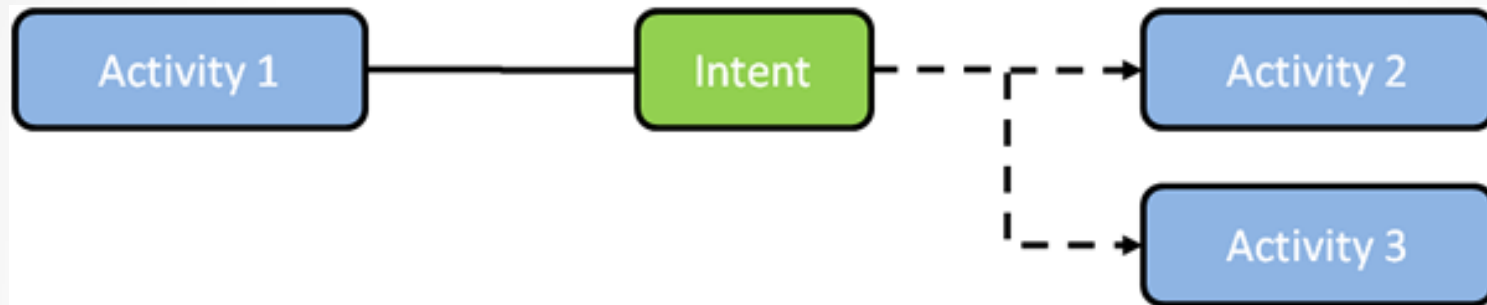
Activity

```
package org.teste;  
import android.app.Activity;  
import android.os.Bundle;  
public class TesteActivity extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```

Intent

- Intent (Intenção) indica que a aplicação tem a intenção de realizar algo. Cada ação representa uma Intent (uma mudança de tela, por exemplo). Você pode usar as intenções para realizar as seguintes tarefas:
 - Transmitir uma mensagem.
 - Início de um serviço.
 - Carregando uma Atividade.
 - Mostrar uma página web ou uma lista de contatos.
 - Discar um número de celular ou atender uma chamada telefônica.

Modelo orientado a serviço



- As atividades podem declarar quais ações são capazes de responder
- Carregamento de uma atividade via Intent pode ser
 - Explícito; especificando qual classe deve ser aberta o
 - Implícito; especificando o “serviço” desejado

Carregamento Implícito

- Serviço de View URI:

```
Intent intent = new  
Intent(Intent.ACTION_VIEW);  
intent.setData(  
    Uri.parse("http://www.google.com"));  
this.startActivity(intent);
```

Enviando uma mensagem

```
Intent intent = new  
Intent(Intent.ACTION_SEND);  
intent.setType("text/plain");  
intent.putExtra(android.content.Intent.EXTRA_  
TEXT, "vamo dormir meu povo!");  
startActivity(Intent.createChooser(intent,  
"Escolhe aí!: "));
```

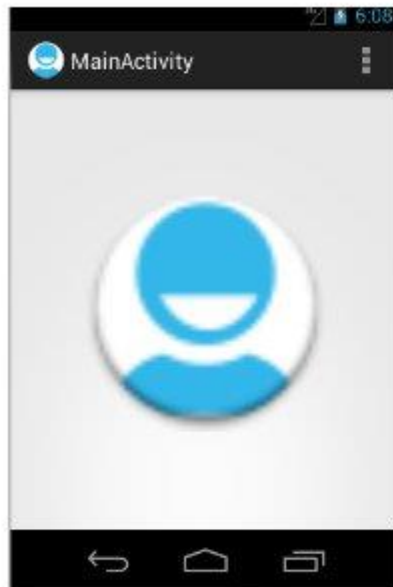

Exemplos de Tipos de Actions

- Olhadinha em:
- <http://www.vogella.de/articles/AndroidIntent/article.html>

SubActivity

```
Intent intent = new  
Intent(Intent.ACTION_VIEW);  
intent.setData(  
    Uri.parse("http://www.google.com"));  
activity.startActivityForResult(intent);
```

SubActivity



Intent resolution by the
Android system



One activity is
started

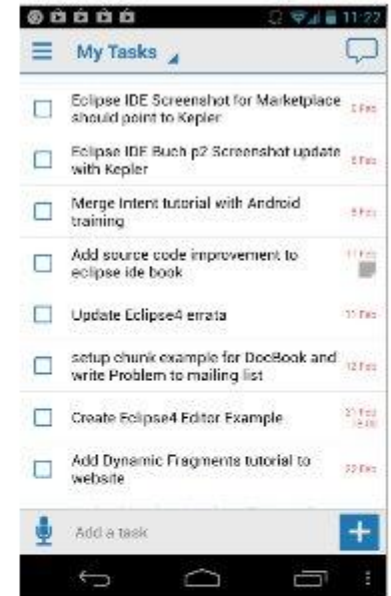
SubActivity



`onActivityResult(requestCode, resultCode, intent)`



Intent + resultCode
provided by called
activity



requestCode
provided by Android to
identify which activity
type was started

Outro Exemplo

- Intent intent = **new**
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
- `// fileUri = getOutputMediaFileUri(MEDIA_TYPE_IMAGE); //`
create a file to save the image
- `// intent.putExtra(MediaStore.EXTRA_OUTPUT, fileUri); // set`
the image file name
- `// start the image capture Intent`
- `startActivityForResult(intent,`
`CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE);`

Exercício

- Se o login e a senha forem equivalentes as valores “a” e “b” abram o navegador na página da ufc virtual

Passando parâmetros

- Antes de invocar o método startActivity use
- A função putExtra
- Intent intent= **new Intent(this, Atividade2.class);**
- intent.putExtra("param1",et.getText().toString());
- startActivity(intent);
- Pode passar tipos básicos e objetos serializáveis

Recuperando parâmetros

- Use o método `getExtras` para recuperar uma instância de `Bundle` com todos os parâmetros passados.
- `Intent it=this getIntent();`
- `Bundle params=it.getExtras();`
- `String param1= params.getString("param1");`

Para funcionar...

- Toda Activity de um projeto deve ser registrada no Android Manifest
- `<activity`
- `android:label="@string/app_name"`
- `android:name=".MyBooksActivity" >`
- `</activity>`