



INSTITUTO FEDERAL  
PARANÁ

# DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS

PROF<sup>a</sup>. M.Sc. JULIANA H Q BENACCHIO



# Retornando Informações

- O método **startActivity()** chama outra activity
- Se esta nova activity for finalizada, a activity que fez a chamada não recebe nenhuma informação



# Retornando Informações

- Para notificar a activity chamadora, é possível usar **`startActivityForResult()`**
- A nova activity carregada é considerada uma sub-activity, já que fica atrelada à activity que a chamou



# Retornando Informações

**Tela1.java**

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    Intent it = new Intent(getApplicationContext(), Tela2.class);  
    startActivityForResult(it, 1);  
}  
  
protected void onActivityResult(int requestCode, int resultCode,  
    Intent data) {  
  
    //...  
}
```

É preciso fornecer, além da intent, um código de identificação

Invocado quando a subactivity chamada é finalizada



# Retornando Informações

**Tela2.java**

```
setResult(10);  
finish();
```

Define um código de resposta

Finaliza a activity

Tela1

```
onActivityResult(1, 10, null)
```



# Passagem de Parâmetros

- É possível passar parâmetros de uma activity para outra usando um **Bundle**

## Tela1.java

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    Intent it = new Intent(getApplicationContext(), Tela2.class);  
    Bundle params = new Bundle();  
    params.putString("msg", "Alguma mensagem");  
    it.putExtras(params);  
    startActivity(it);  
}
```



# Passagem de Parâmetros

- A leitura do parâmetro é feita a partir da recuperação da **intent** que disparou a **activity**

Tela2.java

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    Intent it = getIntent();  
    Bundle params = it.getExtras();  
    String msg = params.getString("msg");  
}
```



# Salvando o Estado de uma Activity



- Uma **activity** pode ser destruída pelo Android para liberar recursos
- Quando ela for recriada, é possível que informações da **activity** sejam perdidas
- Para evitar que isto aconteça, é necessário salvar o estado da **activity**



# Salvando o Estado de uma Activity



- Não é preciso se preocupar com o estado das **views**, pois ele é gravado e depois recuperado
  - É preciso que a **view** tenha um **ID** especificado



# Método `onSaveInstanceState()`



- O Android chama o método `onSaveInstanceState()` na **activity** quando ela não fica mais visível

```
public class MyActivity extends Activity {  
    private String nome;  
    private int idade;  
  
    protected void onSaveInstanceState(Bundle outState) {  
        outState.putString("nome", nome);  
        outState.putInt("idade", idade);  
        //...  
        super.onSaveInstanceState(outState);  
    }  
}
```

Bundle para  
salvar os dados

Chama o método  
da superclasse

- Quando a **activity** é destruída, o estado dela fica guardado



# Restaurando o Estado de uma Activity



- Quando a **activity** for recriada, o Android chama o método **onCreate()** passando como parâmetro o **bundle** que contém os dados salvos

Se o bundle não é null, a activity está sendo restaurada

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    if (savedInstanceState != null) {  
        this.nome = savedInstanceState.getString("nome");  
        this.idade = savedInstanceState.getInt("idade");  
    }  
}
```

- O método **onRestoreInstanceState(Bundle)** também é chamado (depois de **onStart()**)

