



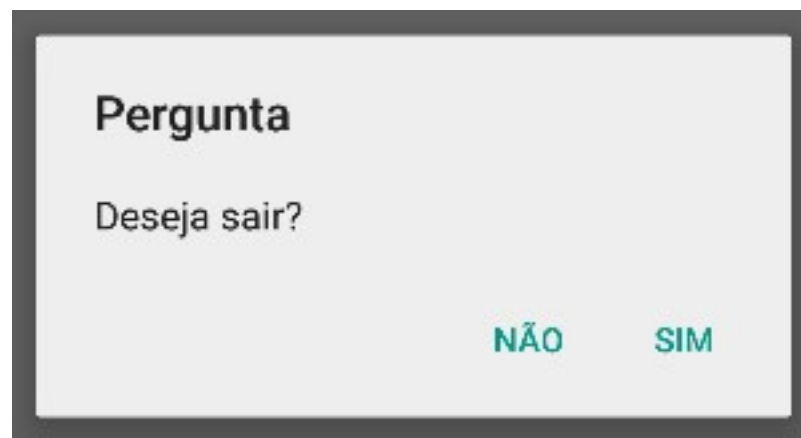
# DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS

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



# Dialogs

- **Dialogs** são caixas de informações que servem para alertar o usuário sobre algo ou para solicitar algum tipo de informação
- Duas classes principais
  - **AlertDialog**: Representa um dialog
  - **AlertDialog.Builder**: Constroi um dialog



# A Classe **DialogFragment**

- A forma recomendada de exibir **dialogs** é através do uso de um **DialogFragment**
- Permite o correto gerenciamento do ciclo de vida do **dialog**
- Possibilita o reaproveitamento do **dialog** em várias partes da aplicação
- A classe **DialogFragment** surgiu na versão 3.0 do Android (API Level 11)



# Criando um Dialog

```
public class MyDialog extends DialogFragment
    implements DialogInterface.OnClickListener {

    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setTitle("Até logo!")
            .setMessage("Deseja sair agora?")
            .setPositiveButton(R.string.sim, this)
            .setNegativeButton(R.string.nao, this);
        return builder.create();
    }

    public void onClick(DialogInterface dialog, int which) {
        if (which == Dialog.BUTTON_NEGATIVE) {
            //...
        } else if (which == Dialog.BUTTON_POSITIVE) {
            //...
        }
    }
}
```

# Exibindo um Dialog

- A exibição é feita pela activity através da instanciação do **dialog** e da chamado ao método **show()**

```
MyDialogFragment dialog = new MyDialogFragment();  
dialog.show(getFragmentManager(), "dialog");
```

Identificador do  
fragment (tag)

FragmentManager



# Criando Dialogs com Listas

- Um **dialog** pode ter uma lista associada
- Pode ser de três tipos
  - Lista Simples
  - Lista Simples com Radio Buttons
  - Lista Múltipla com Checkboxes



# Lista Simples

```
public Dialog onCreateDialog(Bundle savedInstanceState) {  
  
    AlertDialog.Builder builder = new AlertDialog  
        .Builder(getActivity());  
  
    builder.setTitle("Escolha")  
        .setItems(R.array.eletronicos, this);  
  
    return builder.create();  
}  
  
public void onClick(DialogInterface dialog, int which) {  
    //...  
}
```

which armazena o índice  
do elemento escolhido



# Lista Simples com Radio Buttons



```
public Dialog onCreateDialog(Bundle savedInstanceState) {  
  
    AlertDialog.Builder builder = new AlertDialog  
        .Builder(getActivity());  
  
    builder.setTitle("Escolha")  
        .setSingleChoiceItems(R.array.eletronicos, -1, this);  
  
    return builder.create();  
}  
  
public void onClick(DialogInterface dialog, int which) {  
    //...  
}
```

which armazena o índice  
do elemento escolhido





# Lista Múltipla com Checkboxes



```
public Dialog onCreateDialog(Bundle savedInstanceState) {  
    AlertDialog.Builder builder = new AlertDialog  
        .Builder(getActivity());  
  
    builder.setTitle("Escolha")  
        .setMultiChoiceItems(R.array.eletronicos, null, this);  
  
    return builder.create();  
}  
  
public void onClick(DialogInterface dialog, int which,  
    boolean isChecked) {  
    //...  
}
```

*DialogInterface.  
OnMultiChoiceClickListener*

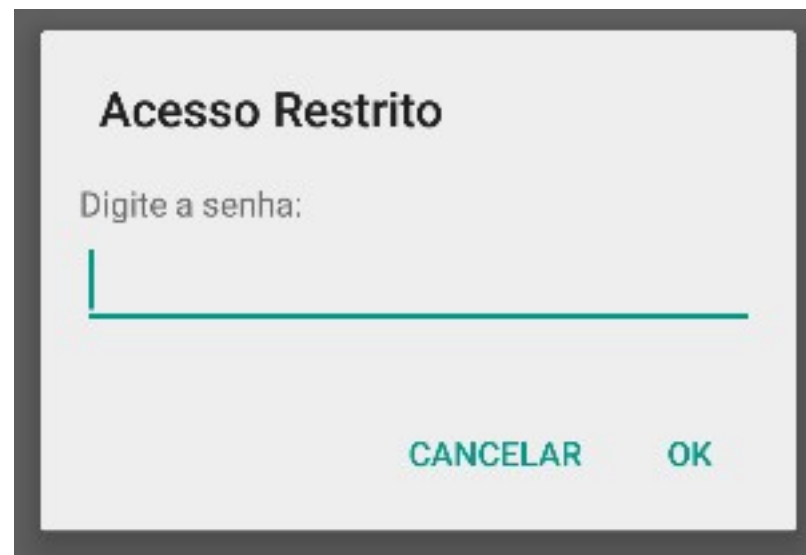
which armazena o índice  
do elemento clicado

isChecked indica se o  
item está selecionado



# AlertDialogs Customizados

- Para uma maior flexibilidade sobre a aparência de um **AlertDialog**, é possível definir uma **view** a ser aplicada ao **dialog**



# AlertDialogs Customizados

/res/layout/dialog.xml

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/txt_senha" />

    <EditText
        android:id="@+id/edt_senha"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberPassword" />

</LinearLayout>
```



# AlertDialogs Customizados

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);  
builder.setTitle("Acesso Restrito");
```

```
LayoutInflater inflater = (LayoutInflater) getActivity().getLayoutInflater();  
View dialogView = inflater.inflate(R.layout.pwd_dialog, null);  
builder.setView(dialogView);
```

```
builder.setPositiveButton(R.string.ok, this);  
builder.setNegativeButton(R.string.cancelar, this);
```

```
...
```

