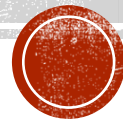


# ANDROID

Caixas de Diálogo



## Caixas de Diálogo

- Link: <https://developer.android.com/guide/topics/ui/dialogs.html?hl=pt-br>
- Classe base : Dialog
- Mas pode-se usar classes filhas pré-definidas
  - **AlertDialog**: caixa básica com título, até 3 botões, lista de itens selecionáveis ou um layout personalizado
  - **DatePickerDialog**: caixa para seleção de data
  - **TimePickerDialog**: caixa para seleção de hora





# AlertDialog

## AlertDialog

- Link: <https://developer.android.com/reference/android/app/AlertDialog.html>
- Para criá-la usa-se um Builder, depois AlertDialog

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);  
// configura a dialog (mensagens e botões)  
AlertDialog dialog = builder.create();  
dialog.show();
```

- Usado para setar as características da Dialog
  - Mensagem
  - Título
  - Ações de botões
- Link: <https://developer.android.com/reference/android/app/AlertDialog.Builder.html>



## AlertDialog

- Métodos do Builder:
  - Todos retornam o próprio Builder, portanto podem ser encadeados
  - `setMessage()`: seta a mensagem da dialog
  - `setTitle()`: seta o título da dialog
  - `setPositiveButton()`: seta o texto (`R.string.id`) e um `DialogInterface.OnClickListener` para interceptar a ação do usuário ao pressionar o botão de aceite
  - `setNegativeButton()`: seta o texto (`R.string.id`) e um `DialogInterface.OnClickListener` para interceptar a ação do usuário ao pressionar o botão de cancelamento da ação
  - `setNeutralButton()`: seta o texto (`R.string.id`) e um `DialogInterface.OnClickListener` para interceptar a ação do usuário ao pressionar o botão para parar a ação, mas não cancelá-la
  - `create()`: cria (mas não mostra) a dialog e retorna um objeto do tipo `AlertDialog`



## AlertDialog — setPositiveButton()

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);

// Seta a mensagem e o título
builder.setMessage(R.string.dialog_message)
    .setTitle(R.string.dialog_title);

// Seta o tratador do pressionamento do botão
builder.setPositiveButton(R.string.ok,
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            // DialogInterface dialog: a dialog que recebeu o clique
            // int id : o botão que foi clicado, que pode ser:
            //             DialogInterface.BUTTON_POSITIVE
            //             DialogInterface.BUTTON_NEGATIVE
            //             DialogInterface.BUTTON_NEUTRAL
        }
    });
```



## AlertDialog – Criação

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setMessage(R.string.dialog_message)
    .setTitle(R.string.dialog_title);
builder.setPositiveButton(R.string.ok,
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            // O que fazer quando o usuário aperta o SIM/OK
        }
    });
builder.setNegativeButton(R.string.cancel,
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            // O que fazer quando o usuário aperta o NÃO/CANCELAR
        }
    });
AlertDialog dialog = builder.create();
dialog.show();
```



## AlertDialog – Layout – activity\_main.xml

```
<RelativeLayout 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="br.net.razer.testedialog1.MainActivity">

    <TextView
        android:id="@+id/texto"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Teste de Dialog" />

    <Button
        android:id="@+id/botao"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/texto"
        android:text="Mostrar Dialog"/>

</RelativeLayout>
```



## AlertDialog – MainActivity.java

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Button btn = (Button) findViewById(R.id.botao);
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            mostrarDialog1();
        }
    });
}
```



## AlertDialog – MainActivity.java

```
public void mostrarDialog1() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage(R.string.dialog_message)
        .setTitle(R.string.dialog_title);
    builder.setPositiveButton(R.string.ok,
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                Toast.makeText(getApplicationContext(), "Apertou SIM",
                    Toast.LENGTH_SHORT).show();
            }
        });
    builder.setNegativeButton(R.string.cancel,
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                Toast.makeText(getApplicationContext(), "Apertou NÃO",
                    Toast.LENGTH_SHORT).show();
            }
        });
    AlertDialog dialog = builder.create();
    dialog.show();
}
```



## Exercício

- Implementar a aplicação anterior e observar o comportamento da Dialog



## DatePickerDialog

## DatePickerDialog

- Abre uma Dialog para escolha de Data
- Link: <https://developer.android.com/reference/android/app/DatePickerDialog.html>
- Construtor
  - Recebe o contexto, `OnDateSetListener`, data atual (ano, mês e dia)
  - Método `show()` para mostrar a dialog
- `DatePickerDialog.OnDateSetListener`
  - Evento para tratar quando a data foi selecionada
  - Método invocado:

```
public void onDateSet(DatePicker view,
                      int year,
                      int month,
                      int dayOfMonth) {

    // faz o que tem que fazer
}
```



## DatePickerDialog

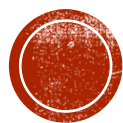
```
public void mostrarDialogData() {
    Calendar agora = Calendar.getInstance();
    int ano = agora.get(Calendar.YEAR);
    int mes = agora.get(Calendar.MONTH);
    int dia = agora.get(Calendar.DAY_OF_MONTH);
    DatePickerDialog dateDialog = new DatePickerDialog(this,
        new DatePickerDialog.OnDateSetListener() {
            public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
                Calendar c = Calendar.getInstance();
                c.set(year, month, dayOfMonth);
                Date dataSelecionada = c.getTime();
                TextView t = (TextView) findViewById(R.id.texto);
                t.setText ((new SimpleDateFormat("dd/MM/yyyy")).format(dataSelecionada));
            }
        },
        ano, mes, dia);
    dateDialog.show();
}
```



## DatePickerDialog

- Para abrir a dialog, pode-se adicionar um botão com evento OnClick

```
botaoDate = (Button)findViewById(R.id.botaoDate);  
botaoDate.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        mostrarDialogData();  
    }  
});
```



## TimePickerDialog



## TimePickerDialog

- Abre uma Dialog para escolha de Hora
- Link: <https://developer.android.com/reference/android/app/TimePickerDialog.html>
- Construtor
  - Recebe o contexto, `OnTimeSetListener`, hora atual (hora e minuto) e um booleano indicando se é representação de 24h
  - Método `show()` para mostrar a dialog
- `TimePickerDialog.OnTimeSetListener`
  - Evento para tratar quando a hora foi selecionada
  - Método invocado:

```
public void onTimeSet(DatePicker view,
                      int hourOfDay,
                      int minute) {
    // faz o que tem que fazer
}
```



## DatePickerDialog

```
public void mostrarDialogHora() {
    Calendar agora = Calendar.getInstance();
    TimePickerDialog timeDialog = new TimePickerDialog(this,
        new TimePickerDialog.OnTimeSetListener() {
            @Override
            public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
                Calendar c = Calendar.getInstance();
                c.set(Calendar.HOUR_OF_DAY, hourOfDay);
                c.set(Calendar.MINUTE, minute);
                Date dataSelecionada = c.getTime();
                TextView t = (TextView)findViewById(R.id.texto);
                t.setText ((new SimpleDateFormat("HH:mm")).format(dataSelecionada));
            }
        },
        agora.get(Calendar.HOUR_OF_DAY), agora.get(Calendar.MINUTE), true);
    timeDialog.show();
}
```



## TimePickerDialog

- Para abrir a dialog, pode-se adicionar um botão com evento onClick

```
botaoTime = (Button)findViewById(R.id.botaoTime);  
botaoTime.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        mostrarDialogTime();  
    }  
});
```

