

Curso Android para *Iniciantes*



Lana Mesquita
Cristiane Mayara
Brunno Melo

Contatos



- Lana Mesquita
lanabeatriz.mesquita@gmail.com
- Cristiane Mayara
cristiane.mayara@gmail.com
- Brunno Melo
brunnomelo@gmail.com
- Grupo Facebook:
<https://www.facebook.com/groups/CursoAndroidCAET2014>

Ementa de hoje

1. LogCat
2. Toast
3. Eventos
 - Input Events
 - TextView
 - EditText
 - Button
4. Qualifiers
5. Prática



LogCat

Existe 5 tipos de logs são eles:

V — Verbose

D — Debug

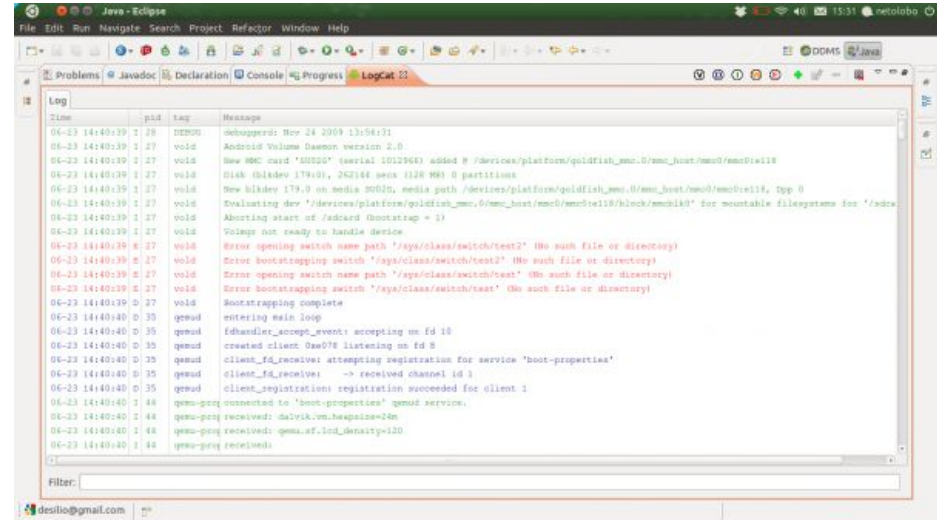
I — Info

W — Warning

E — Error

Para abrir a view do LogCat:

menu Window -> Show View -> Other -> Android -> LogCat



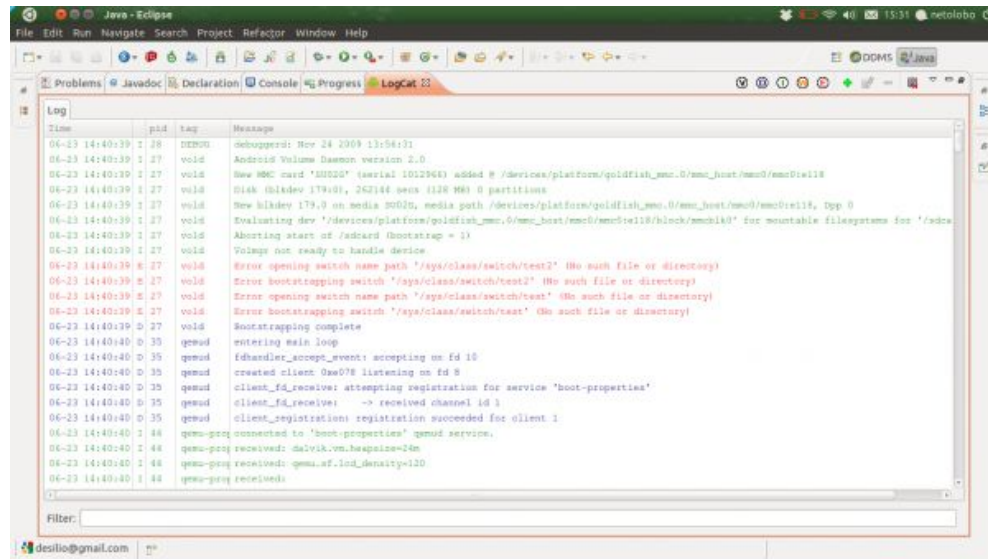
LogCat

A view contém todos os logs do Android, incluindo:

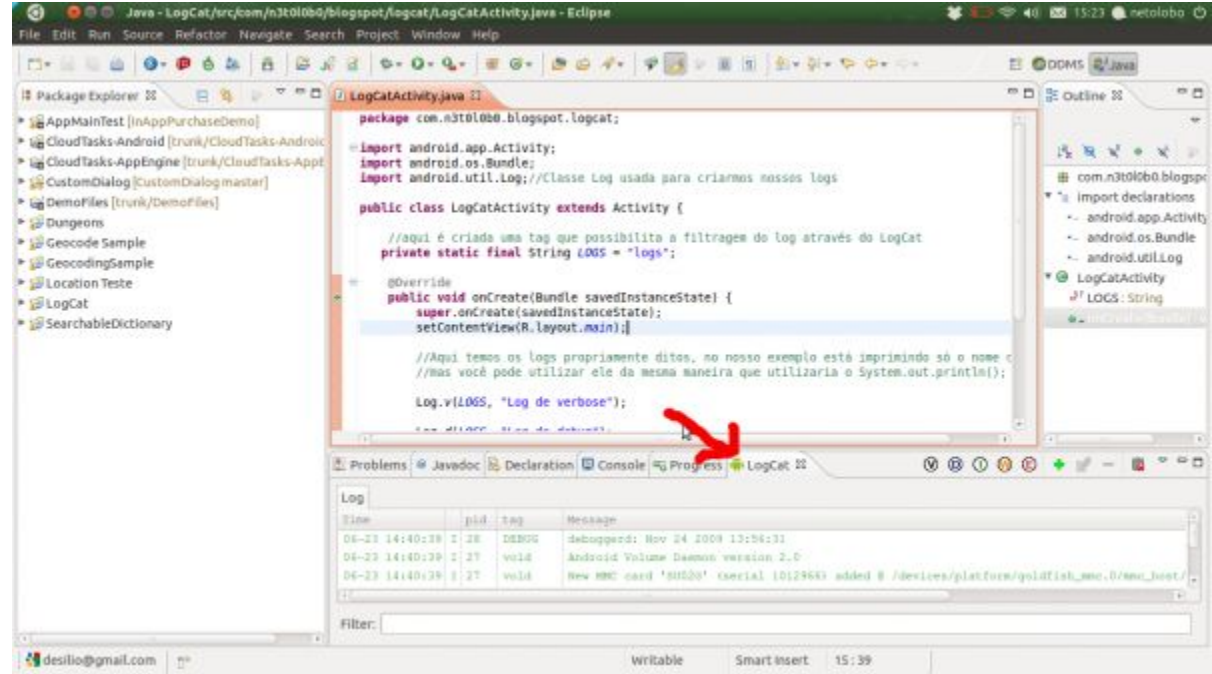
- os logs criados em código
- os logs do sistema

É possível filtrar quais logs deseja ver, por tipo, tag, projeto etc.

Métodos chamados em código para criar um log: `Log.v()` `Log.d()` `Log.i()` `Log.w()` and `Log.e()`



LogCat



- Prática: Insira o método abaixo no método onCreate de uma Activity.
Log.i("teste_info", "Teste do LogCat!");

LogCat

Boas práticas:

Log.e : Indicam erro. Pode ser usado, por exemplo, dentro de um tratamento de um *catch*.

Log.w : Utilizado para indicar um comportamento inesperado, porém não indica com certeza um erro.

Log.i : Utilizado para postar informações úteis para o log. Por exemplo : se você conectou com sucesso a um servidor.

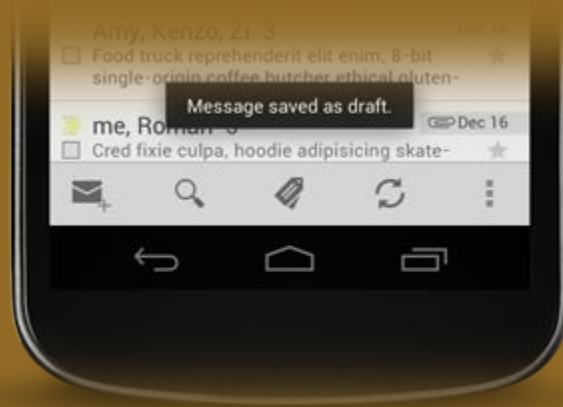
Log.d : Utilizado para fins de depuração. Se você quiser imprimir um monte de mensagens para que você possa registrar o fluxo exato de seu programa, use este. Se você quiser manter um registro de valores de variáveis , use este .

Log.v : Utilizado para registrar detalhes de uma parte específica do seu aplicativo. Como eventos que ocorrem.

Log.wtf : Utilizado para reportar uma falha que nunca deveria acontecer.

Toast

- Toast é uma mensagem de notificação que é exibida um determinado período de tempo, e automaticamente desaparece.
- Pode ser usado para fins de depuração.



Toast

Exemplo:

```
Context context = getApplicationContext();  
CharSequence text = "Hello toast!";  
int duration = Toast.LENGTH_SHORT;  
  
Toast toast = Toast.makeText(context, text, duration);  
toast.show();
```

Para setar sua localização na tela:

```
toast.setGravity(Gravity.TOP|Gravity.LEFT, 0, 0);
```

Input Events

- Há **interação do usuário** através da interface de usuário (*UI*).
- Eventos de entrada de um objeto específico são capturados e precisam ser **tratados na classe java**.
- Métodos *callback* são chamados pelo *framework* do Android quando a ação ocorre sobre o respectivo objeto.
- Para interceptar deve-se implementar a classe e sobrescrever o tal método.



Input Events

- Métodos *callback*

onClick()

É chamado quando o usuário clica no item (quando em *touch mode*), ou foca sobre o item com as teclas de navegação.

onLongClick()

É chamado quando o usuário clica e segura o item (quando em *touch mode*), ou foca sobre o item com as teclas de navegação.

onFocusChange()

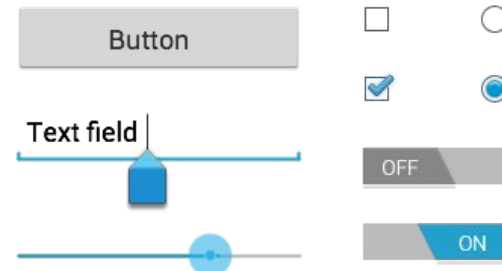
É chamado quando o usuário navega para fora do item.

onKey()

É chamado quando o usuário foca no item e pressiona uma tecla do dispositivo.

onTouch()

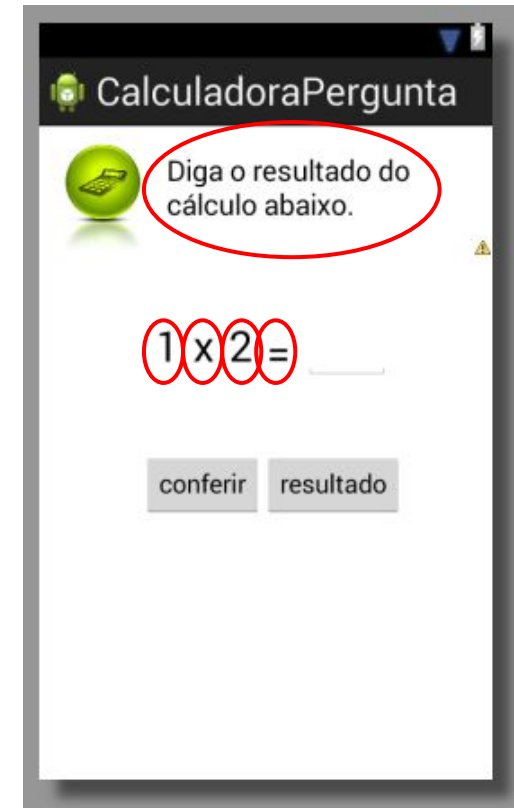
É chamado quando o usuário faz um toque específico, como movimentos de desbloqueio.



TextView

No xml:

```
<TextView  
    android:id="@+id/my_text_view"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_margin="10sp"  
    android:text="@string/main_text"  
    android:textAlignment="center"  
    android:textColor="#FFFFFF"  
    android:textSize="20sp" />
```



TextView

Na classe java:

```
// Instanciar textview
```

```
TextView textview;
```

```
textview = (TextView) findViewById(R.id.my_text_view);
```

```
// Setar texto no textview
```

```
textview.setText("My string.");
```

```
// Setar a fonte no textview
```

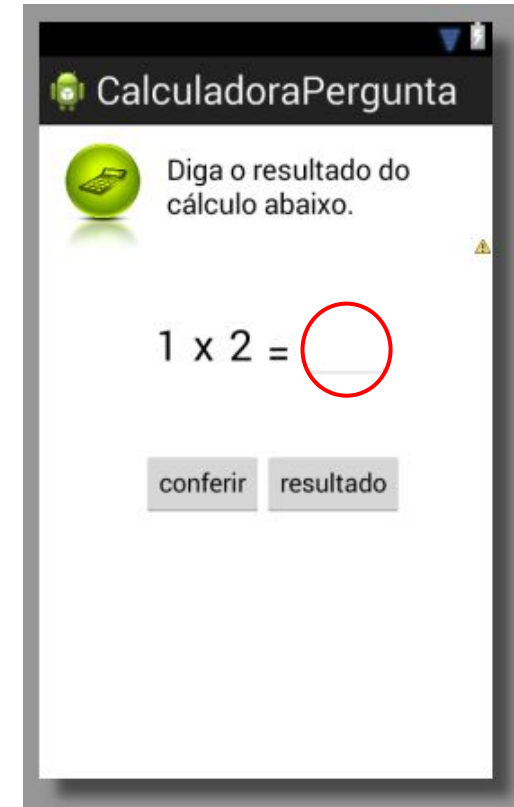
```
Typeface type = Typeface.createFromAsset(getAssets(), "Channel.ttf");
```

```
textview.setTypeface(type);
```

EditText

No xml:

```
<EditText  
    android:id="@+id/my_edit_text"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_margin="10sp"  
    android:cursorVisible="true"  
    android:inputType="number"  
    android:textColor="#FFFFFF"  
    android:textSize="20sp" />
```



EditText

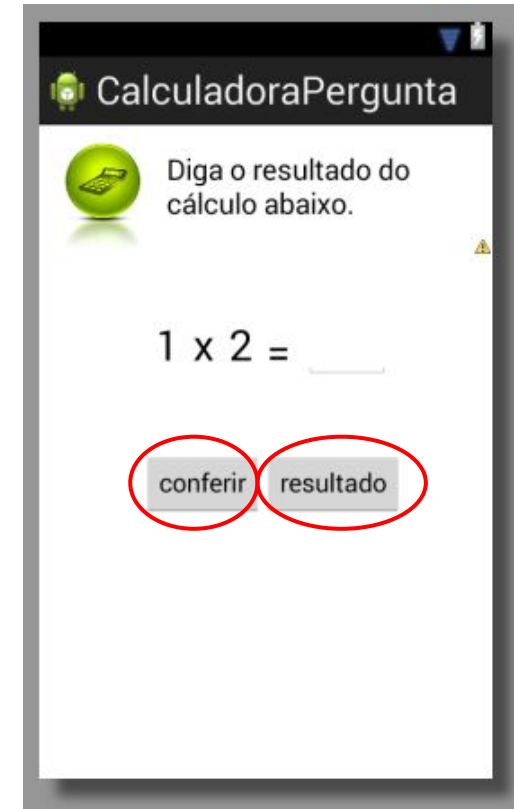
Na classe java:

```
// Instanciar edittext  
EditText edittext;  
  
edittext = (EditText) findViewById(R.id.my_edit_text);  
  
// Obter string do edittext  
String str = edittext.getText().toString();  
  
// Converter string para inteiro  
int int_value = Integer.parseInt(str);
```

Button

No xml:

```
<Button  
    android:id="@+id/my_button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_margin="10sp"  
    android:background="#0099CC"  
    android:padding="5dp"  
    android:textColor="#FFFFFF"  
    android:textSize="20sp" />
```



Button

Na classe java:

```
// Instanciar button

Button button;

button = (Button) findViewById(R.id.my_button);

// Tratar evento de clique no button

button.setOnClickListener(new OnClickListener() {

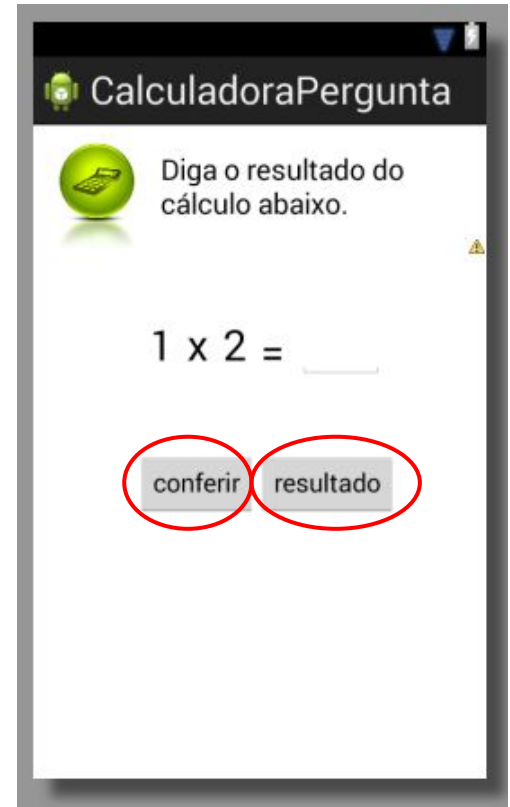
    @Override

    public void onClick(View v) {

        // Inserir o código aqui.

    }

});
```



Qualifiers

- Quase todas as aplicações devem fornecer recursos alternativos para suportar configurações de dispositivos específicos.
- Por exemplo, você deve incluir recursos drawable alternativos para diferentes densidades de tela e recursos de cadeia alternativos para diferentes idiomas.
- Em tempo de execução, o Android detecta a configuração atual do dispositivo e carrega os recursos apropriados para sua aplicação.



Qualifiers

Para criar uma pasta com qualifier na pasta

`res/` chamar da seguinte forma:

`<resources_name>-<config_qualifier>`

`<resources_name>` é a pasta corresponde ao recurso default.

`<qualifier>` é um nome que especifica uma configuração específica.

Podem ser aplicados mais de um `<qualifier>` separados por - que devem seguir uma ordem listada no site developer do Android.

`res/`

`drawable/`

`icon.png`

`background.png`

`drawable-hdpi/`

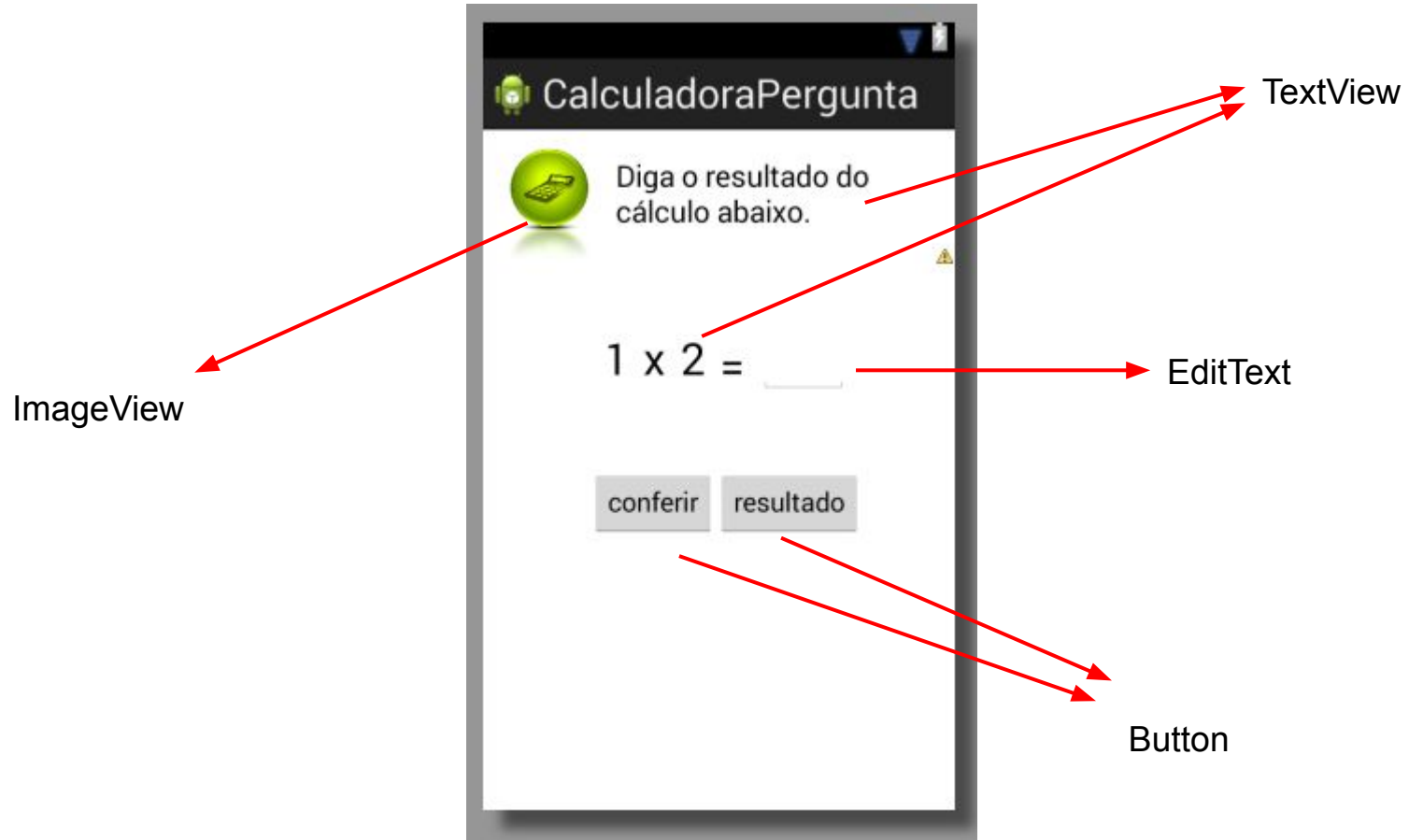
`icon.png`

`background.png`

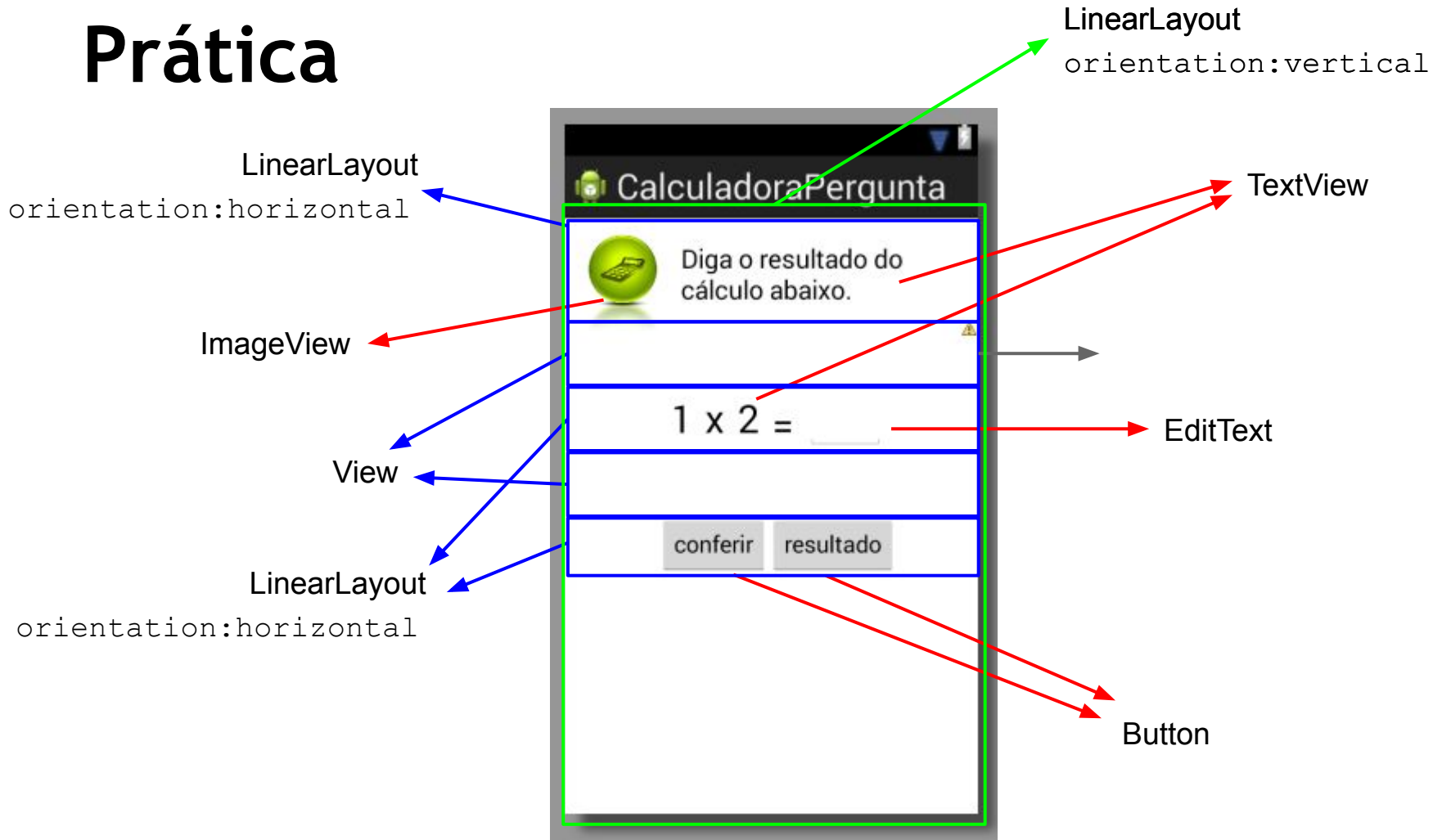
On Windows

- "If you are developing on Windows and would like to connect an Android-powered device to test your applications, then you need to install the appropriate USB driver."
- Procurar o driver OME (*Original Equipment Manufacturers*) do celular:
<http://developer.android.com/tools/extras/oem-usb.html#Drivers>
- Para instalar o driver USB:
<http://developer.android.com/tools/extras/oem-usb.html#Win7>
- Obs.: Tem para Windows 7, Xp e vista.

Prática

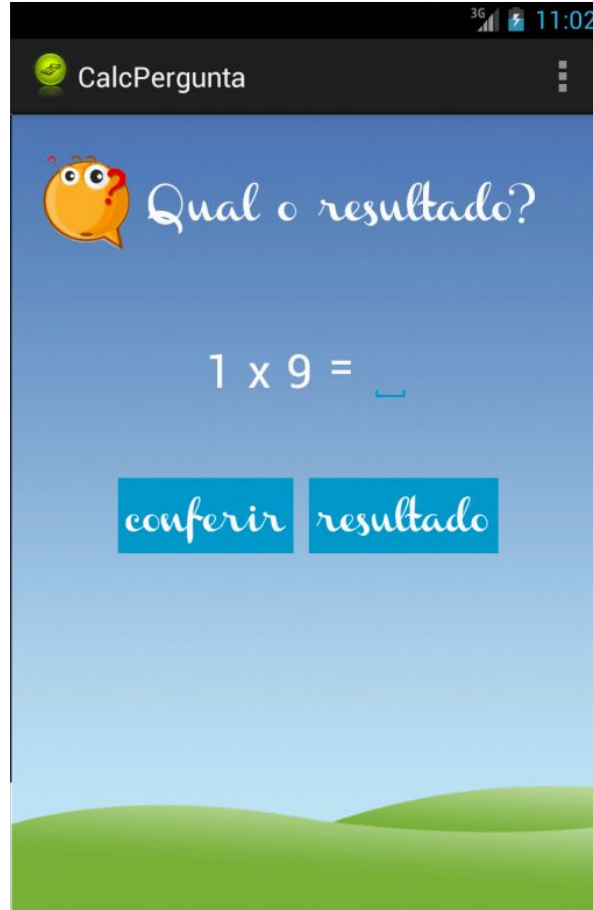


Prática



Prática de hoje

LinearLayout
orientation:vertical



Desafio

Criar dois layouts, um para portrait e outro para landscape!

