

Mensagens

Qual a finalidade da mensagem que é exibida para o usuário?

- Toasts não necessariamente serão lidas pelos usuários pois aparecem e somem rapidamente.
- Alerts são mensagens críticas para o usuário. Mensagens que ele deve ler.

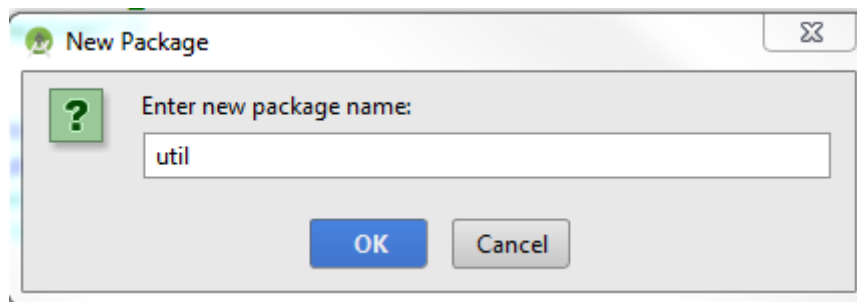
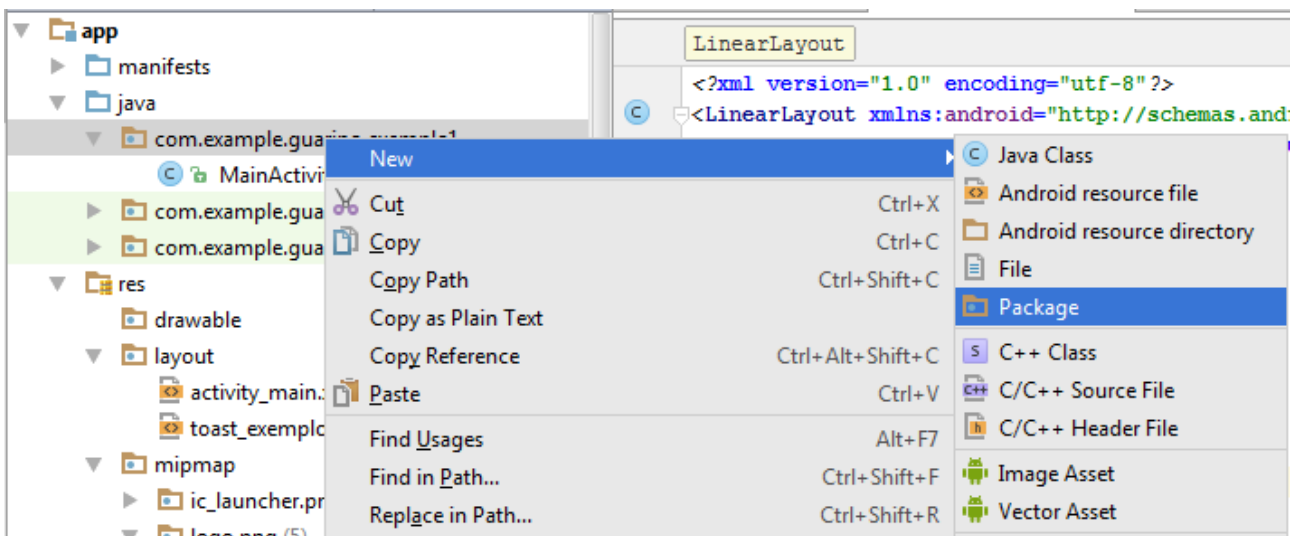
Para conhecimento. Renomear arquivos.

Botão Direito no arquivo > Refactor > Rename...

Shift + F6

Criar uma classe Util para colocar uma função para exibir Toast

Criar novo pacote



Adicionar classe

Adicionar classe Util.java dentro desse pacote

MainActivity.java

Recortar o conteúdo colorido – LayoutInflater até toast.show para o método a ser criado na classe Util.

```
package com.example.guarino.exemplo1;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().setTitle(R.string.app_title);

        //infla o layout - permite adicionar informações ao layout
        LayoutInflater inflater = getLayoutInflater();
        View layout_toast = inflater.inflate(R.layout.toast_exemplo, (ViewGroup)
findViewById(R.id.layout_toast));
        TextView txtToast = (TextView) layout_toast.findViewById(R.id.txtToast);
        txtToast.setText("Exemplo de app 1.0");
        //Mensagem rápida - Toda activity é um context
        //Toast toast = Toast.makeText(MainActivity.this, "Teste de Exemplo app 1.0",
Toast.LENGTH_SHORT);
        Toast toast = new Toast(this);
        toast.setView(layout_toast);
        //set x e y - valores positivos ou negativos
        toast.setGravity(Gravity.CENTER, 0, 0);
        toast.show();
    }
}
```

Criar um método na classe Util.java

Criar método

```
public static void showMessageToast(Activity activity, String texto)
```

Copiar o conteúdo anterior para dentro do método. Ao fazer isso, o Android Studio irá mostrar todas as classes que ele irá importar.

O método deve ficar como segue:

```
package com.example.guarino.exemplo1.util;
import android.app.Activity;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;
import com.example.guarino.exemplo1.R;

public class Util {

public static void showMessageToast(Activity activity, String texto)
{
    LayoutInflater inflater = activity.getLayoutInflater();

    View layout_toast = inflater.inflate(R.layout.toast_exemplo, (ViewGroup)
activity.findViewById(R.id.layout_toast));

    TextView txtToast = (TextView) layout_toast.findViewById(R.id.txtToast);
    txtToast.setText(texto);

    Toast toast = new Toast(activity);
    toast.setView(layout_toast);
    toast.setGravity(Gravity.CENTER, 0, 0);
    toast.show();
}
}
```

Na MainActivity.java, chamar o método criado em Util.

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().setTitle(R.string.app_title);
        Util.showMessageToast(this, "Exemplo de Toast");
    }
}
```

Executar a app no AVD - Será exibida uma mensagem no centro da tela.

Criar um alert

@Override

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

    getSupportActionBar().setTitle(R.string.app_title);

    Util.showMessageToast(this, "Exemplo de Toast");

    AlertDialog alert = new AlertDialog.Builder(this).create();
    alert.setTitle("Exemplo alert");
    alert.setMessage("Mensagem de alert");
    alert.setIcon(R.mipmap.logo);
    alert.show();
}
```

Executar a app no AVD

Será exibida uma mensagem toast e um alert.

Criar um alert com botão OK

@Override

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

    getSupportActionBar().setTitle(R.string.app_title);

    Util.showMessageToast(this, "Exemplo de Toast");

    AlertDialog alert = new AlertDialog.Builder(this).create();
    alert.setTitle("Exemplo alert");
    alert.setMessage("Mensagem de alert");
    alert.setIcon(R.mipmap.logo);

    alert.setButton(DialogInterface.BUTTON_POSITIVE, "OK", new
    DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
        }
    });
    alert.show();
}
```

Executar a app no AVD

Será exibida uma mensagem toast e um alert.

Criar um alert com Cancelar e OK

@Override

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

    getSupportActionBar().setTitle(R.string.app_title);

    Util.showMessageToast(this, "Exemplo de Toast");

    AlertDialog alert = new AlertDialog.Builder(this).create();
    alert.setTitle("Exemplo alert");
    alert.setMessage("Mensagem de alert");
    alert.setIcon(R.mipmap.logo);

    alert.setButton(DialogInterface.BUTTON_POSITIVE, "OK", new
    DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
        }
    });

    alert.setButton(DialogInterface.BUTTON_NEGATIVE, "Cancelar", new
    DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
        }
    });

    alert.show();
}
```

Executar a app no AVD

Será exibida uma mensagem toast e um alert.

Criando um alert com OK para ser reutilizado

Na classe Util.java, criar o método

```
public static void showMessageOK (Activity activity, String titulo, String texto)
{
    AlertDialog alert = new AlertDialog.Builder(activity).create();
    alert.setTitle(titulo);
    alert.setMessage(texto);
    alert.setIcon(R.mipmap.logo);
    alert.setButton(DialogInterface.BUTTON_POSITIVE, "OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
        }
    });
    alert.show();
}
```

Na classe MainActivity.java, fazer a chamada do método

```
Util.showMessageOK(this, "Exemplo de alert", "Exemplo de alert OK");
```

Executar a app no AVD

Será exibida uma mensagem toast e um alert.

Alterando a largura e altura do Alert

Na classe Util.java, alterar o método showMessageOK

```
public static void showMessageOK (Activity activity, String titulo, String texto)
{
    AlertDialog alert = new AlertDialog.Builder(activity).create();
    alert.setTitle(titulo);
    alert.setMessage(texto);
    alert.setIcon(R.mipmap.logo);
    alert.setButton(DialogInterface.BUTTON_POSITIVE, "OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
        }
    });

    WindowManager.LayoutParams params = new WindowManager.LayoutParams();

    //Largura do tamanho do Layout e altura do tamanho do texto
    params.copyFrom(alert.getWindow().getAttributes());
    params.width = WindowManager.LayoutParams.MATCH_PARENT;
    params.height = WindowManager.LayoutParams.WRAP_CONTENT;

    alert.show();

    alert.getWindow().setAttributes(params);
    alert.getWindow().setBackgroundDrawableResource (R.color.verdeEscuro);

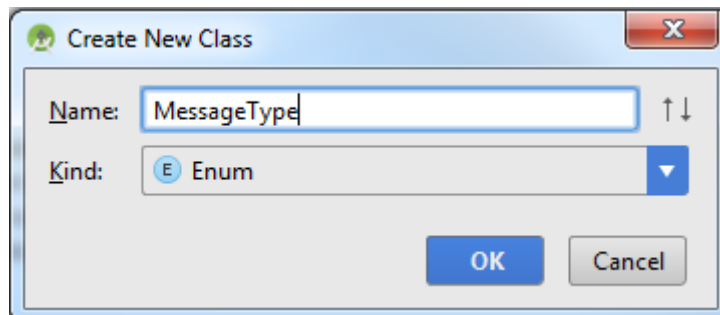
}
```

Executar a app no AVD

Será exibida uma mensagem toast e um alert.

Criando um enum para definir o tipo de mensagem

Na pasta `java/com.example.guarino.exemplo1/util` > New > Java Class

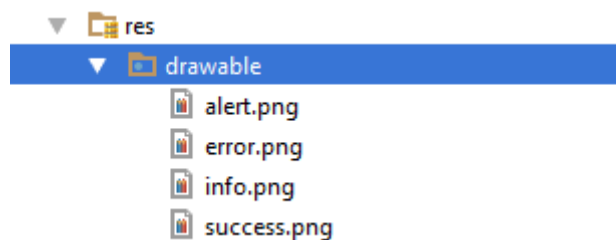


Obs: Alterar o Kind para Enum

package com.example.guarino.exemplo1.util;

```
public enum MessageType {  
    ERRO, INFO, SUCESSO, ALERTA  
}
```

Adicionar imagens na pasta `res/drawable` para os 4 tipos de mensagens. Para isso, basta dar Ctrl+C nos arquivos no Windows Explorer e Ctrl+V na pasta citada.



Alterar o método showMessageOK em Util.java

```
public static void showMessageOK (Activity activity, String titulo, String texto,
MessageType tipo)
{
    AlertDialog alert = new AlertDialog.Builder(activity).create();
    alert.setTitle(titulo);
    alert.setMessage(texto);

    switch (tipo){
        case INFO:
            alert.setIcon(R.drawable.info);
            break;
        case ERRO:
            alert.setIcon(R.drawable.error);
            break;
        case SUCCESSO:
            alert.setIcon(R.drawable.success);
            break;
        case ALERTA:
            alert.setIcon(R.drawable.alert);
            break;
    }
    //alert.setIcon(R.mipmap.logo);

    alert.setButton(DialogInterface.BUTTON_POSITIVE, "OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
        }
    });
    WindowManager.LayoutParams params = new WindowManager.LayoutParams();
    params.copyFrom(alert.getWindow().getAttributes());
    params.width = WindowManager.LayoutParams.MATCH_PARENT;
    params.height = WindowManager.LayoutParams.WRAP_CONTENT;
    alert.show();
    alert.getWindow().setAttributes(params);
    //alert.getWindow().setBackgroundDrawableResource(R.color.verdeEscuro);
}
}
```

Alterar a chamada do método em MainActivity.java

```
Util.showMessageOK(this, "Exemplo de alert", "Exemplo de alert OK",
MessageType.SUCCESSO);
```

Executar a app no AVD

Será exibida uma mensagem toast e um alert.

Alterar a cor de cada tipo de mensagem

No arquivo res/values/colors.xml
Criar as cores abaixo:

```
<?xml version="1.0" encoding="utf-8" ?>
<resources>
  <color name="colorPrimary">#3F51B5</color>
  <color name="colorPrimaryDark">#303F9F</color>
  <color name="colorAccent">#FF4081</color>
  <color name="branco">#FFFFFF</color>
  <color name="verdeEscuro">#004106</color>
  <color name="azulClaro">#4bb2e2</color>
  <color name="vermelho">#ff0000</color>
  <color name="amarelo">#eaff00</color>
</resources>
```

No arquivo res/values/styles.xml
Criar os estilos abaixo:

```
<resources>
  <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
  </style>
  <style name="AppTheme.Dark.Dialog.Alert" parent="Theme.AppCompat.Dialog">
    <item name="colorAccent">@color/branco</item>
    <item name="android:textColorPrimary">@color/branco</item>
    <item name="android:background">@color/verdeEscuro</item>
  </style>
  <style name="AppTheme.Dark.Dialog.Info" parent="Theme.AppCompat.Dialog">
    <item name="colorAccent">@color/branco</item>
    <item name="android:textColorPrimary">@color/branco</item>
    <item name="android:background">@color/amarelo</item>
  </style>
  <style name="AppTheme.Dark.Dialog.Success"
parent="Theme.AppCompat.Dialog">
    <item name="colorAccent">@color/branco</item>
    <item name="android:textColorPrimary">@color/branco</item>
    <item name="android:background">@color/azulClaro</item>
  </style>
  <style name="AppTheme.Dark.Dialog.Error" parent="Theme.AppCompat.Dialog">
    <item name="colorAccent">@color/branco</item>
    <item name="android:textColorPrimary">@color/branco</item>
    <item name="android:background">@color/vermelho</item>
  </style>
</resources>
```

Alterar o método showMessageOK na classe Util.java

```
public static void showMessageOK (Activity activity, String titulo, String texto,
MessageType tipo)
{
    int theme = 0;
    int icon = 0;
    switch (tipo){
        case INFO:
            icon = R.drawable.info;
            theme = R.style.AppTheme_Dark_Dialog_Info;
            break;
        case ERRO:
            icon = R.drawable.error;
            theme = R.style.AppTheme_Dark_Dialog_Error;
            break;
        case SUCESSO:
            icon = R.drawable.success;
            theme = R.style.AppTheme_Dark_Dialog_Success;
            break;
        case ALERTA:
            icon = R.drawable.alert;
            theme = R.style.AppTheme_Dark_Dialog_Alert;
            break;
    }
    AlertDialog alert = new AlertDialog.Builder(activity, theme).create();
    alert.setTitle(titulo);
    alert.setMessage(texto);
    alert.setIcon(icon);
    //alert.setIcon(R.mipmap.logo);

    alert.setButton(DialogInterface.BUTTON_POSITIVE, "OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
        }
    });
    WindowManager.LayoutParams params = new WindowManager.LayoutParams();
    params.copyFrom(alert.getWindow().getAttributes());
    params.width = WindowManager.LayoutParams.MATCH_PARENT;
    params.height = WindowManager.LayoutParams.WRAP_CONTENT;
    alert.show();
    alert.getWindow().setAttributes(params);
}
```

Executar a app no AVD

Será exibida uma mensagem toast e um alert.

Adicionando um botão na MainActivity

No arquivo res/values/strings.xml

```
<resources>
  <string name="app_name">Exemplo1</string>
  <string name="app_title">Exemplo1 - Guarino</string>
  <string name="btnEnviar">Enviar</string>
</resources>
```

No arquivo res/values/dimens.xml

```
<resources>
  <dimen name="activity_horizontal_margin">16dp</dimen>
  <dimen name="activity_vertical_margin">16dp</dimen>
  <dimen name="margin_widgets">20dp</dimen>
  <dimen name="botao">16sp</dimen>
</resources>
```

No arquivo res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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="com.example.guarino.exemplo1.MainActivity">

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />

  <Button
    android:id="@+id/btnEnviar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/btnEnviar"
    android:background="@color/colorPrimaryDark"
    android:textColor="@color/branco"
    android:textSize="@dimen/botao"
    android:layout_alignParentBottom="true"/>

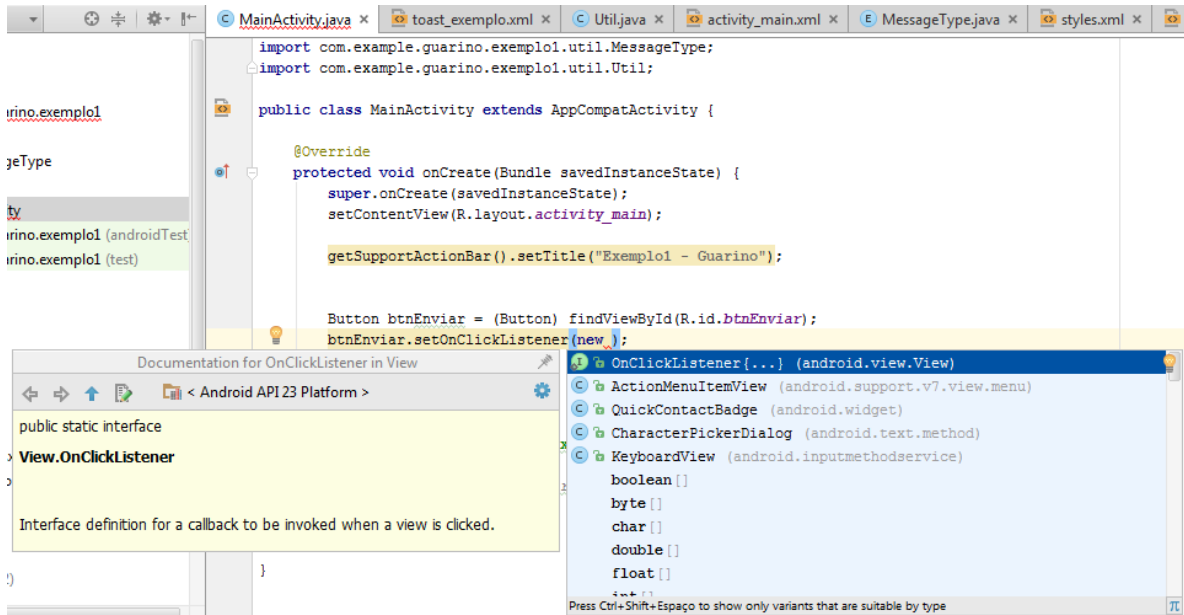
</RelativeLayout>
```

Na MainActivity.java

Adicionar uma chamada para o botão

```
Button btnEnviar = (Button) findViewById(R.id.btnEnviar);  
btnEnviar.setOnClickListener(new CTRL+ESPAÇO);
```

Ao pressionar CTRL+ESPAÇO, irá mostrar a tela a seguir.



Usar o OnClickListener... Deixar o código da MainActivity como:

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        getSupportActionBar().setTitle(R.string.app_title);  
  
        Button btnEnviar = (Button) findViewById(R.id.btnEnviar);  
  
        btnEnviar.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                Util.showMessageOK(MainActivity.this, "Exemplo de alert", "Exemplo de alert OK", MessageType.SUCCESSO);  
            }  
        });  
  
        Util.showMessageToast(this, "Exemplo de Toast");  
    }  
}
```

Executar a app no AVD

Será exibida uma mensagem toast.
Clicar no botão Enviar. Será exibido o Alert.

Definindo uma ação no click do Alert

Mainactivity.java

Remover a chamada do Toast e colocá-la dentro do click do método showMessageOK da classe Util.java

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().setTitle(R.string.app_title);

        Button btnEnviar = (Button) findViewById(R.id.btnEnviar);

        btnEnviar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Util.showMessageOK(MainActivity.this, "Exemplo de alert", "Exemplo de
alert OK", MessageType.SUCESSO);
            }
        });

        //Removido
        //Util.showToast(this, "Exemplo de Toast");
    }
}
```

Util.java

Deixar o parâmetro activity como final.

Deixar a variável alert como final.

Acrescentar o código dentro do click do alert para chamar o Toast.

alert.dismiss() - garante que o Dialog será eliminado.

```
public static void showMessageOK (final Activity activity, String titulo, String texto,
MessageTypes tipo)
```

```
{
    int theme = 0;
    int icon = 0;
    switch (tipo){
        case INFO:
            icon = R.drawable.info;
            theme = R.style.AppTheme_Dark_Dialog_Info;
            break;
        case ERRO:
            icon = R.drawable.error;
            theme = R.style.AppTheme_Dark_Dialog_Error;
            break;
        case SUCESSO:
            icon = R.drawable.success;
            theme = R.style.AppTheme_Dark_Dialog_Success;
            break;
        case ALERTA:
            icon = R.drawable.alert;
            theme = R.style.AppTheme_Dark_Dialog_Alert;
            break;
    }
}
```

```
final AlertDialog alert = new AlertDialog.Builder(activity, theme).create();
```

```
    alert.setTitle(titulo);
    alert.setMessage(texto);
    alert.setIcon(icon);
```

```
    alert.setButton(DialogInterface.BUTTON_POSITIVE, "OK", new
DialogInterface.OnClickListener() {
```

```
        @Override
```

```
        public void onClick(DialogInterface dialog, int which) {
```

```
            Util.showMessageToast(activity, "Exemplo de Toast");
```

```
            //garante o fechamento do dialog
```

```
            alert.dismiss();
```

```
        }
```

```
    });
```

```
WindowManager.LayoutParams params = new WindowManager.LayoutParams();
params.copyFrom(alert.getWindow().getAttributes());
```

```

params.width = WindowManager.LayoutParams.MATCH_PARENT;
params.height = WindowManager.LayoutParams.WRAP_CONTENT;

alert.show();
alert.getWindow().setAttributes(params);
}

```

Executar a app no AVD

Clicar no botão Enviar. Será exibido o Alert. Ao clicar no OK, será exibido um Toast.

Uma segunda forma de executar o onClick do botão

Alterar a MainActivity

- Remover a chamada do evento click
- Criar uma function clickBtnEnviar

```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().setTitle(R.string.app_title);

//REMOVED
        Button btnEnviar = (Button) findViewById(R.id.btnEnviar);
        btnEnviar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Util.showMessageOK(MainActivity.this, "Exemplo de alert", "Exemplo de
alert OK", MessageType.SUCESSO);
            }
        });
    }

    public void clickBtnEnviar(View view)
    {
        Util.showMessageOK(MainActivity.this, "Exemplo de alert", "Exemplo de alert
OK", MessageType.SUCESSO);
    }
}

```


Alterar a activity_main.xml

Colocar a chamada do onClick

```
<Button
    android:id="@+id/btnEnviar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/btnEnviar"
    android:background="@color/colorPrimaryDark"
    android:textColor="@color/branco"
    android:textSize="@dimen/botao"
    android:onClick="clickBtnEnviar"
    android:layout_alignParentBottom="true"/>
```

Executar a app no AVD

Clicar no botão Enviar. Será exibido o Alert. Ao clicar no OK, será exibido um Toast.

Criar um novo método em Util.java com 2 botões (OK e Cancelar)

Método semelhante ao showMessageOK mas com mais um botão (linhas amarelas abaixo).

Alterar o conteúdo do click OK para apenas imprimir no Log (uso de Log.d).

```
public static void showMessageOKCancelar (final Activity activity, String titulo, String texto,
    MessageType tipo)
```

```
{
    int theme = 0;
    int icon = 0;

    switch (tipo){
        case INFO:
            icon = R.drawable.info;
            theme = R.style.AppTheme_Dark_Dialog_Info;
            break;
        case ERRO:
            icon = R.drawable.error;
            theme = R.style.AppTheme_Dark_Dialog_Error;
            break;
        case SUCESSO:
            icon = R.drawable.success;
            theme = R.style.AppTheme_Dark_Dialog_Success;
            break;
        case ALERTA:
            icon = R.drawable.alert;
            theme = R.style.AppTheme_Dark_Dialog_Alert;
            break;
    }
}
```

```
final AlertDialog alert = new AlertDialog.Builder(activity, theme).create();
alert.setTitle(titulo);
alert.setMessage(texto);
alert.setIcon(icon);
```

```

alert.setButton(DialogInterface.BUTTON_POSITIVE, "OK", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        //imprime -1
        Log.d("OK", String.valueOf(which));
        //garante o fechamento do dialog
        alert.dismiss();
    }
});

```

```

alert.setButton(DialogInterface.BUTTON_NEGATIVE, "Cancelar", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        //imprime -2
        Log.d("CANCELAR", String.valueOf(which));
        //garante o fechamento do dialog
        alert.dismiss();
    }
});

```

```

WindowManager.LayoutParams params = new WindowManager.LayoutParams();
params.copyFrom(alert.getWindow().getAttributes());
params.width = WindowManager.LayoutParams.MATCH_PARENT;
params.height = WindowManager.LayoutParams.WRAP_CONTENT;
alert.show();
alert.getWindow().setAttributes(params);
//alert.getWindow().setBackgroundDrawableResource(R.color.verdeEscuro);
}

```

Na MainActivity.java, colocar uma chamada para o método, apenas para teste.

```

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

    getSupportActionBar().setTitle(R.string.app_title);

    Util.showMessageOKCancelar(MainActivity.this, "Exemplo de alert", "Exemplo de
alert OK", MessageType.SUCESSO);
}

```

Executar a app no AVD

Será exibido o Alert. Clicar no OK ou no Cancelar.
Observar no Android Studio, no Android Monitor, o valor da variável impressa.