

1 Ventás de Diálogos 2

1.1 DialogFragment

Estes apuntes están postos como resposta a unha pregunta do foro.

A tarefa de PLATEGA non ten que facerse utilizando este tipo de diálogo.

1.1.1 Introducción

A partires da API 13 (Android 3.2) a forma de 'construír' os diálogos modificouse.

Se usades a forma anteriormente explicada funcionará pero vos marcará en Eclipse a liña como **deprecated**.

```
public void onBotonClick(View view) {  
    switch (view.getId()) {  
        case R.id.btn_dialogo:  
            showDialog(DIALOGO_MENSAXE);  
            break;  
    }  
}
```

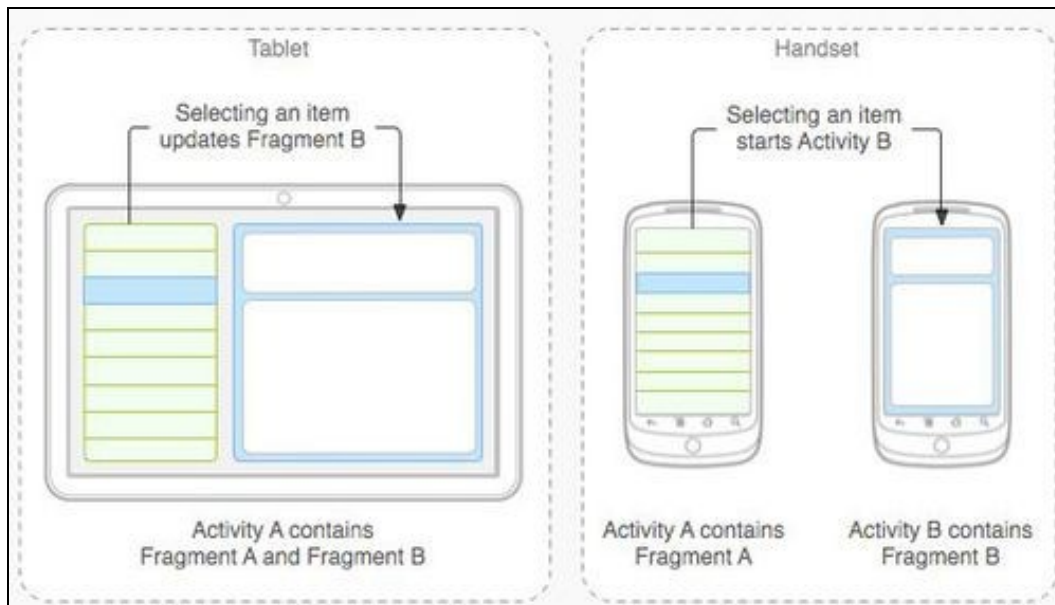
Imos ver neste punto como poderíamos facer para construír os diálogos doutra forma.

Para explicalo teremos que falar antes dos DialogFragment e por extensión dos Fragment.

Información adicional:

- DialogFragment
- Fragment

Un fragment representa un 'trozo' da interface dun usuario. A idea dos fragment xurdiu polas dificultades que tiñan as aplicacións a adaptarse a tamaños grandes de pantalla (como as tablet's).



Imaxe obtida de [1]

Imaxínade que tedes unha aplicación que amosa unha lista de usuarios e que ó premer sobre un deles apareceran os seus datos completos.

Isto feito nun móbil cunha pantalla de 4 polgadas levaría consigo a creación de dúas actividades, xa que na mesma non cabe toda a información (necesitaríamos usar scroll, bastante incómodo).

Esta mesma aplicación levada a unha tablet de 10 polgadas non tería problema en visualizar todo na mesma pantalla.

Os fragment vannon permitir dividir os elementos gráficos en 'anacos' que imos poder usar de forma independente nas activities. Así, no caso anterior podemos crear dous fragment, un coa lista e outro có detalle de cada usuario.

Agora dependendo do tamaño da pantalla pode facer que na pantalla pequena se amose un só fragment (a lista) é que o premer sobre un elemento da mesma apareza o outro fragment (o detalle).

Se o tamaño é grande (tablet) pode facer que aparezan os dous fragment na mesma activity.

O código que xestiona a pulsación dun elemento da lista e amosar os datos estará definido no só sitio, o que modificamos é o sitio onde se visualiza o fragment en función do tamaño da pantalla.

Tendo claro o concepto de Fragment isto nos leva ó seguinte concepto: **DialogFragment**.

1.1.2 DialogFragment

- Un DialogFragment é un Fragment que amosa unha ventá de diálogo e sitúase por enriba da nosa Activity.



Os DialogFragment poden ser utilizados en versións anteriores á API 13, pero nese caso teremos que utilizar unha [biblioteca de compatibilidade](#) que nos proporciona Eclipse:



Polo tanto se temos un Min SDK no android manifesto menor á API 13 teremos que importar a clase DialogFragment de dita librería:

```
import android.support.v4.app.DialogFragment;
```

Un DialogFragment vai ter un layout asociado que vai compoñer o seu contido.

Definimos polo tanto o contido do noso diálogo:

Código da clase fragment_u3_15_layout_dialogo

```
<?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" >

        <EditText
            android:id="@+id/editTexto"
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:inputType="text"
            android:gravity="fill_horizontal"
        />
        <Button
            android:id="@+id/buttonPecharDialogo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/editTexto"
            android:layout_centerInParent="true"
            android:text="Pechar" />

    </RelativeLayout>

```

Agora deberemos crear unha clase que vai representar o DialogFragment e que vai cagar o layout anterior. Esta clase será a que instanciemos dende a nosa Activity.

Código da clase UD3_15_DialogoFragmento

Obxectivo: Clase que representa ó DialogFragment

```

import android.os.Bundle;
import android.support.v4.app.DialogFragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;

public class UD3_15_DialogoFragmento extends DialogFragment{

    public String valorTexto;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

        final View rootView = inflater.inflate(R.layout.fragment_u3_15_layout_dialogo, container, false);
        getDialog().setTitle(getTag()); // O Tag se envía dende a activiy có método show.

        Button btn = (Button) rootView.findViewById(R.id.buttonPecharDialogo);
        btn.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                EditText edit = (EditText)rootView.findViewById(R.id.editTexto);
                valorTexto = edit.getText().toString();
                ((U3_15_Dialogos)UD3_15_DialogoFragmento.this.getActivity()).pecharDialogo();
                dismiss();
            }
        });

        // Do something else
        return rootView;
    }
}

```

Analicemos o código.

- Liña 2: Estamos a utilizar a librería de compatibilidade con versións anteriores de Android.
- Liña 10: A nosa clase deriva de DialogFragment.
- Liña 15: Ó derivar de DialogFragment sobreescribimos o método onCreateView que ten que devolver o View que conforma o diálogo.

No noso caso deberemos 'inflar' o layout deseñado anteriormente.

O obxecto da clase `LayoutInflater` permite pasar dun deseño gráfico (o layout) e un obxecto manexable por programación. Isto o fai chamando ó método `inflate` e o garda no obxecto `rootView`.

- Liña 21: Có paso anterior, podemos acceder ós compoñentes gráficos que se atopan no layout por programación. No noso caso accedemos ó botón pechar e xestionamos o evento de click.
- Liñas 25-31: O que facemos nestas liñas é acceder ó contido da caixa de texto do diálogo, o gardamos nunha propiedade da clase.

◊ Liña 28: Esta liña vos dará un erro ata que useses o código que ven a continuación. Esta liña chama a un método definido na `activity` que lanzou o `DialogFragment`.

◊ Liña 29: Pechamos o diálogo.

Agora implantaremos a `activity` que vai facer uso do `DialogFragment`.

Aquí teremos dúas opcións:

- Derivar a clase de `Activity`. Deberemos facer uso do método `getFragmentManager()`.
- Derivar a clase de `FragmentActivity`. Esta clase é unha subclase de `Activity` e está posta por compatibilidade coas **versións anteriores**.

No caso de utilizar esta clase deberemos facer uso do método `getSupportFragmentManager`. **No noso exemplo imos facer uso deste método.**

Definimos o layout da nosa `activity`:

Código da clase `activity_u3_15__dialogos.xml`

Obxectivo: Layout da nosa `activity`.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Ventás de diálogo" />

    <Button
        android:id="@+id/btn_dialogo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onBotonClick"
        android:text="Ventá de diálogo con fragment" >
    </Button>

</LinearLayout>
```

E agora o código da `Activity`:

Código da clase `U3_15_Dialogos`

Obxectivo: Amosar un diálogo baseado en `Fragment`.

```
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.FragmentManager;
import android.view.View;
import android.widget.Toast;

public class U3_15_Dialogos extends FragmentActivity {

    private UD3_15_DialogoFragmento dialogoFragmento;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

        setContentView(R.layout.activity_u3_15__dialogos);
        dialogoFragmento = new UD3_15_DialogoFragmento();
    }

    public void pecharDialogo(){
        Toast.makeText(this, dialogoFragmento.valorTexto, Toast.LENGTH_LONG).show();
    }

    public void onBotonClick(View view) {
        FragmentManager fm = getSupportFragmentManager();

        switch (view.getId()) {
            case R.id.btn_dialogo:
                dialogoFragmento.show(fm, "EXEMPLO DE DIALOGO!!!");
                break;
        }
    }
}

```

Analicemos o código.

- Liñas 2,3,7: Por compatibilidade con versións anteriores de Android escollemos a opción de utilizar un FragmentActivity.
- Liña 9,15: Definimos e instaciamos o diálogo.
- Liñas 18-20: Este é o método que chama a clase do Diálogo cando prememos o botón pechar.
- Liña 23: Necesitamos un FragmentManager para amosar o diálogo.
- Liña 27: Amosamos o diálogo.

Esta é unha primeira aproximación os DialogFragment. Comentar que en vez de sobreescribir o método onCreateView do DialogFragment podemos sobreescribir o método onCreateDialog e devolver un diálogo construído por nos (coma un DatePickerDialog, AlertDialog.builder,...) Por exemplo:

Código da clase UD3_15_DialogoFragmento

Obxectivo: Modificamos o código para amosar outra forma de crear un diálogo. **Debemos comentar o código do método onCreateView.**

```

import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.os.Bundle;
import android.support.v4.app.DialogFragment;
import android.widget.Toast;

public class UD3_15_DialogoFragmento extends DialogFragment{

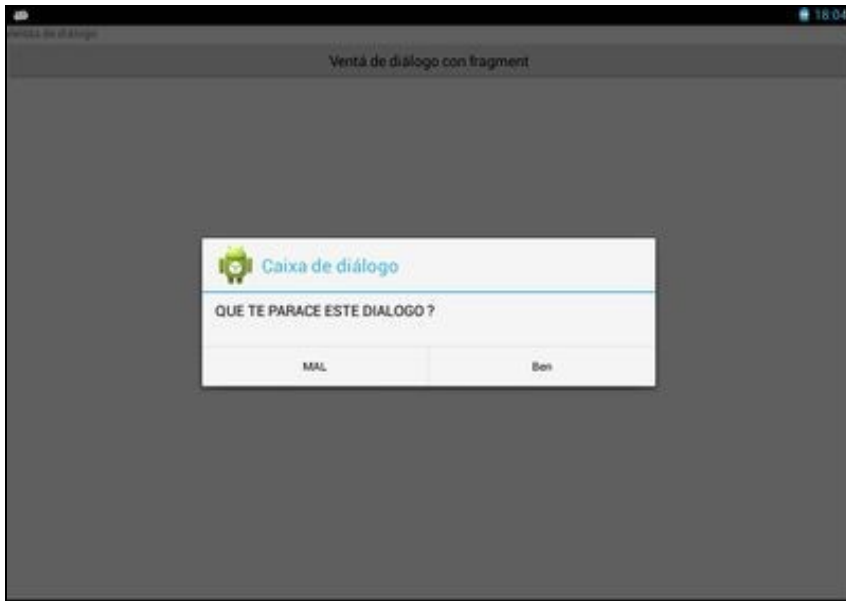
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity())
            .setTitle("Caixa de diálogo").setIcon(R.drawable.ic_launcher)
            .setMessage("QUE TE PARACE ESTE DIALOGO ?")
            .setPositiveButton("Ben", new OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    Toast.makeText(getActivity(), "PULSADA OPCION BOA", Toast.LENGTH_LONG).show();
                }
            }).setNegativeButton("MAL", new OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    //
                    Toast.makeText(getActivity(), "PULSADA OPCION MALA", Toast.LENGTH_LONG).show();
                }
            });
        return builder.create();
    }
}

```

}

}

Nota: Fixarse que estamos a importar a interface onClickListener do DialogInterface (líña 4).



-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2014).