PDM Avanzado Permisos AndroidManifiest.xml

Sumario

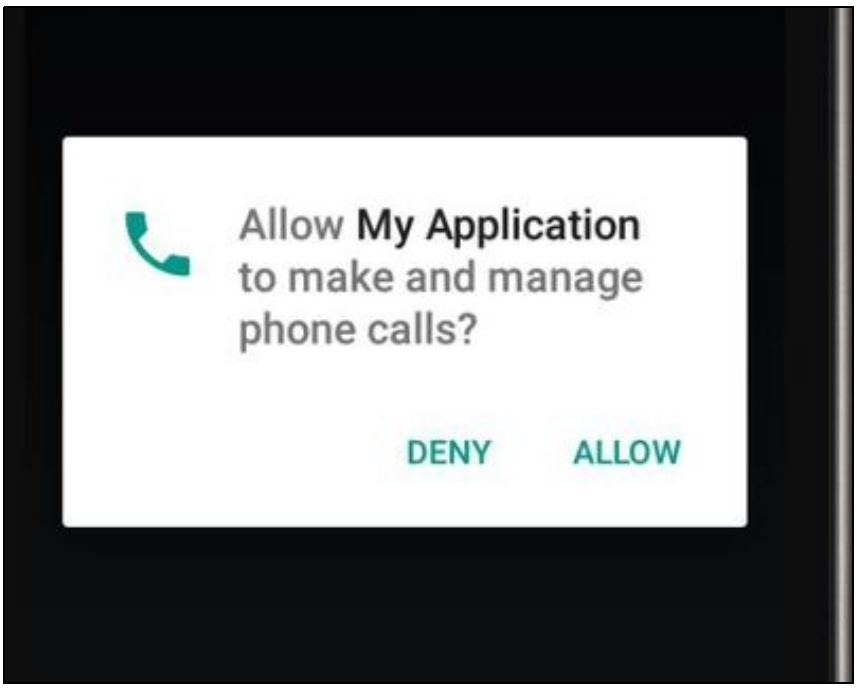
- 1 Introdución
- 2 Tipos de permisos
- 3 Comprobando se temos o permiso
- 4 Solicitando o permiso
- 5 Caso práctico
    - 5.1 Creamos a activity

Introdución

Como xa sabedes cando facemos algunha aplicación que necesite permisos especiais, é necesario engadir ditos permisos ao arquivo AndroidManifiest.xml.

A partires da API 23, Android incorpora outro nivel de seguridade, de tal forma que agora, a maiores, teremos que solicitar dito permiso a nivel de programación.

Nese entre ao usuario apareceralle un diálogo no que se lle solicitará o seu permiso para que a aplicación faga uso do recurso.



Isto só se aplica a permisos denominados **perigosos** (o veremos a continuación) e executando a aplicación nun dispositivo cunha versión Android 6.0 e un targetSDK 23. En caso contrario, o permiso o pedirá como ata o de agora, no momento da instalación.

Máis información en:

- http://developer.android.com/training/permissions/requesting.html
- http://developer.android.com/guide/topics/security/permissions.html

Tipos de permisos

Nota: Isto só é aplicable a aplicacións cun TargetSDK 23 e executadas nun dispositivo cunha API 23 (Android 6.0).

En Android teremos dous tipos de permisos:

- Normais: permisos que non afectan á privacidade do usuario. Estes só teñen que estar incluídos no arquivo androidmanifiest.xml
- Perigosos: permisos que poden afectar a privacidade do usuario e teñen que ser permitidos polo usuario no momento que se necesiten por parte da aplicación.

Temos unha lista dos permisos denominados perigosos neste enlace.

Comprobando se temos o permiso

O permiso non é necesario pedilo sempre. Con tal de que o dea a primeira vez xa queda 'gardado'. Polo tanto podemos chequear se xa temos o permiso do usuario.

Neste exemplo estamos a solicitar un permiso para chamar por teléfono.

Previamente teríamos que ter posto no **androidmanifiest.xml** o permiso correspondente:

```
<uses-permission android:name="android.permission.CALL_PHONE" />
```

O código dentro da activity:

```
if (Build.VERSION.SDK_INT==23){
                    int permiso = checkSelfPermission(Manifest.permission.CALL_PHONE);
                    if (permiso ==PackageManager.PERMISSION_GRANTED){
                            // TEMOS O PERMISO
                    }
                    else{
                            // NON TEMOS O PERMISO TEREMOS QUE SOLICITALO
                    }
                }
```

- Liña 1: Comprobamos que a versión é a API 23. Isto é necesario se temos un minSDK inferior á 23.
- Liña 2: Chamamos ao método checkSelfPermission, que devolve un número que nos vai a indicar se temos o permiso.
- Liña 3: Comprobamos se temos o permiso (granted ou denied).

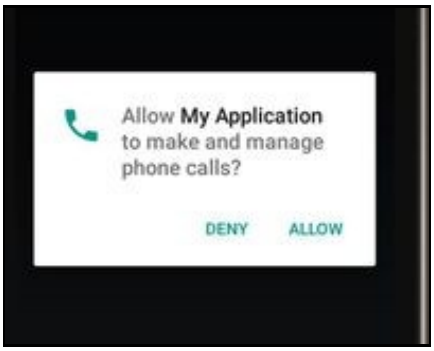
Neste código suponse que a activity deriva da clase Activity e non dunha AppCompatActivity (librerías de compatibilidade). Se é o caso, o código varía un pouco:

```
if (Build.VERSION.SDK_INT==23){
                    int permiso = ContextCompat.checkSelfPermission(this,Manifest.permission.CALL_PHONE);
                    if (permiso ==PackageManager.PERMISSION_GRANTED){
                            // TEMOS O PERMISO
                    }
                    else{
                            // NON TEMOS O PERMISO TEREMOS QUE SOLICITALO
```

```
        }
    }
```

Solicitando o permiso

Para solicitar o permiso temos que chamar a un método que dará como resultado a visualización dunha caixa de diálogo;



O resultado dese diálogo vai ir a o **método onRequestPermissionsResult**. Como podemos ter varios permisos diferentes, podemos enviar un código asociado a dita solicitude para que despois poidamos distinguir cal foi o permiso concedido ou denegado por parte do usuario.

Exemplo de código:

```
// Usado por si necesitamos diferentes permisos, para identificar cual de ellos es
   private final int CODIGO_IDENTIFICADOR=1;

   public void pedirPermiso(){

      requestPermissions( new String[]{Manifest.permission.CALL_PHONE},CODIGO_IDENTIFICADOR);

   }

   @Override
    public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {

        switch (requestCode) {
            case CODIGO_IDENTIFICADOR: {
                // Se o usuario premeou o boton de cancelar o array volve cun null
                if (grantResults.length > 0
                        && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                    // PERMISO CONCEDIDO
                } else {
                    // PERMISO DENEGADO
                    Toast.makeText(this,"É NECESARIO O PERMISO PARA CHAMAR POR TELÉFONO",Toast.LENGTH_LONG).show();
                }
                return;
            }

            // Comprobamos os outros permisos

        }
    }
```

- Liña 6: Solicitamos o permiso enviado un código asociado (valor 1)
- Liña 11: Aquí chega a resposta do usuario.
- Liñas 15-22: Xestionamos que o usuario concedera ou non o permiso.

No caso de utilizar unha biblioteca de compatibilidade, o código sería:

```
    public void pedirPermiso(){
```

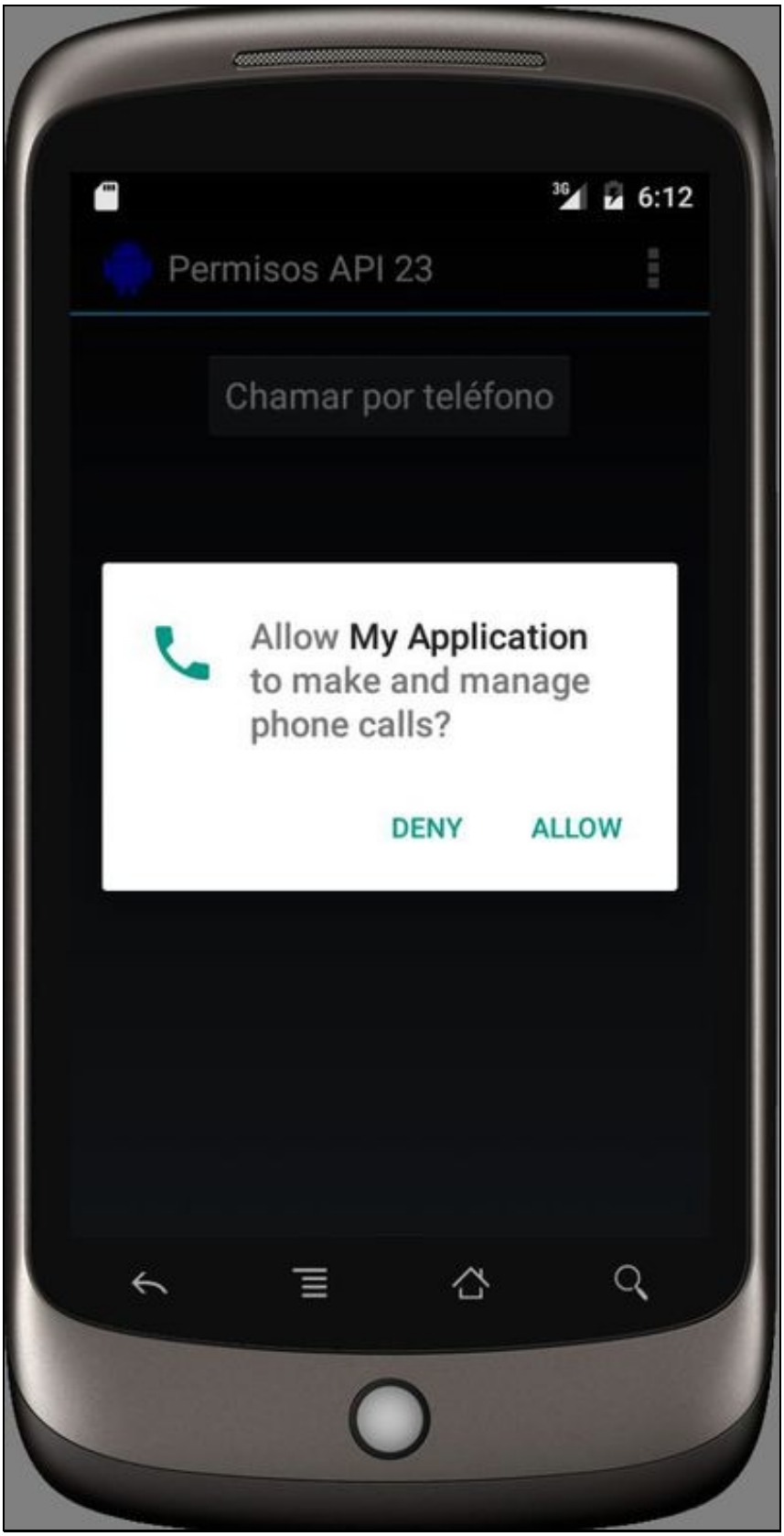
```
            ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.CALL_PHONE},CODIGO_IDENTIFICADOR);

        }
```

Caso práctico

O obxectivo desta práctica e ver pedir un permiso ao S.O. Android de tipo 'perigoso'. Concretamente para chamar a un número de teléfono.

Consta dun botón no que se chamará a un número prefixado.


6:12
Permisos API 23
Chamar por teléfono
Allow My Application to make and manage phone calls?
DENY
ALLOW

Creamos a activity

- Nome do proxecto: **UD08_01_Permisos**
- Nome da activity: **UD08_01_Permisos.java**

Código do layout 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:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="es.cifprodolfoucha.angelfernandez.myapplication.UD08_01_Permisos">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Chamar por teléfono"
        android:id="@+id/UD08_01_btnChamar"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```

Código da clase UD08_01_Permisos

Obxectivo: Solicitar un permiso para un dispositivo Android 6.0 e un targetSDK=23.

Nota: Non se utilizan bibliotecas de compatibilidade.

```
public class UD08_01_Permisos extends Activity {

    // Usado por si necesitamos diferentes permisos, para identificar cual de ellos es
    private final int CODIGO_IDENTIFICADOR=1;

    private void chamarTelefono(){
        Intent callIntent = new Intent(Intent.ACTION_CALL);
        callIntent.setData(Uri.parse("tel:123456789"));
        startActivity(callIntent);
    }

    private void xestionarEventos(){

        Button boton = (Button)findViewById(R.id.UD08_01_btnChamar);
        boton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                if (Build.VERSION.SDK_INT==23){
                    int permiso = checkSelfPermission(Manifest.permission.CALL_PHONE);
                    if (permiso ==PackageManager.PERMISSION_GRANTED){
                        chamarTelefono();
                    }
                    else{
                        UD08_01_Permisos.this.requestPermissions( new String[]{Manifest.permission.CALL_PHONE},CODIGO_IDENTIFICADOR);
                    }

                }
            }
        });


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

```
        setContentView(R.layout.activity_ud08_01__permisos);


        xestionarEventos();

    }

    @Override
    public void onRequestPermissionsResult(int requestCode,
                                           String permissions[], int[] grantResults) {
        switch (requestCode) {
            case CODIGO_IDENTIFICADOR: {
                // If request is cancelled, the result arrays are empty.
                if (grantResults.length > 0
                        && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                    chamarTelefono();
                } else {

                    Toast.makeText(this,"É NECESARIO O PERMISO PARA CHAMAR POR TELÉFONO",Toast.LENGTH_LONG).show();
                }
                return;
            }

            // other 'case' lines to check for other
            // permissions this app might request
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_telefono_carrion, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

-- Ángel D. Fernández González