

# PDM Avanzado Permisos AndroidManifest.xml

---

## Sumario

---

Introdución

Tipos de permisos

Comprobando se temos o permiso

Solicitando o permiso

Caso práctico

Creamos a activity

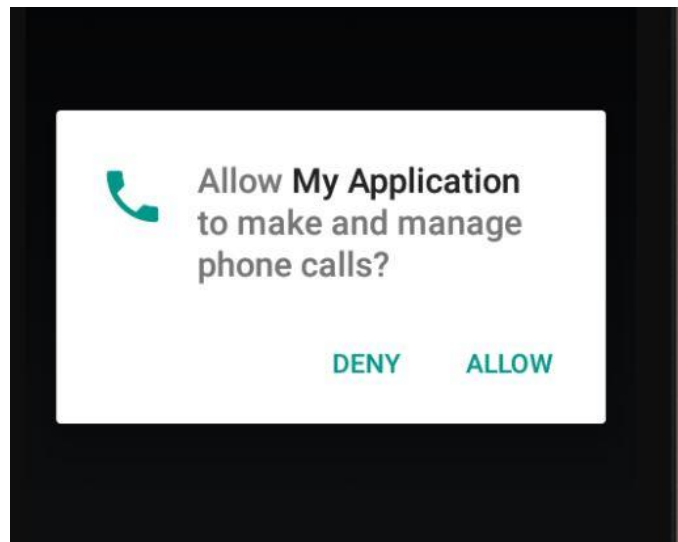
## Introdución

---

Como xa sabedes cando facemos algunha aplicación que necesite permisos especiais, é necesario engadir ditos permisos ao arquivo AndroidManifest.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 androidmanifest.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 (<http://developer.android.com/guide/topics/security/permissions.html#normal-dangerous>).

## 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 **androidmanifest.xml** o permiso correspondente:

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

O código dentro da activity:

```
1 if (Build.VERSION.SDK_INT==23){
2     int permiso = checkSelfPermission(Manifest.permission.CALL_PHONE);
3     if (permiso ==PackageManager.PERMISSION_GRANTED){
4         // TEMOS O PERMISO
5     }
6     else{
7         // NON TEMOS O PERMISO TEREMOS QUE SOLICITALO
8     }
9 }
```

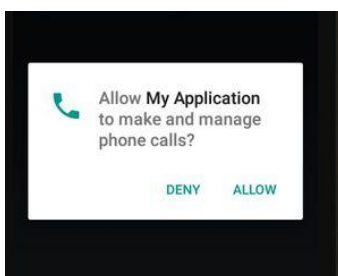
- Liña 1: Comprobamos que a versión é a API 23. Isto é necesario se temos un minSDK inferior á 23.
- Liña 2: Chamamos ao método checkSelfPermission, que devolve un número que nos vai a indicar se temos o permiso.
- Liña 3: Comprobamos se temos o permiso (granted ou denied).

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:

```
1 if (Build.VERSION.SDK_INT==23){
2     int permiso = ContextCompat.checkSelfPermission(this,Manifest.permission.CALL_PHONE);
3     if (permiso ==PackageManager.PERMISSION_GRANTED){
4         // TEMOS O PERMISO
5     }
6     else{
7         // NON TEMOS O PERMISO TEREMOS QUE SOLICITALO
8     }
9 }
```

## 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:

```

1 // Usado por si necesitamos diferentes permisos, para identificar cual de ellos es
2 private final int CODIGO_IDENTIFICADOR=1;
3
4 public void pedirPermiso(){
5
6     requestPermissions( new String[]{Manifest.permission.CALL_PHONE},CODIGO_IDENTIFICADOR);
7
8 }
9
10 @Override
11 public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
12
13     switch (requestCode) {
14         case CODIGO_IDENTIFICADOR: {
15             // Se o usuario premeou o boton de cancelar o array volve cun null
16             if (grantResults.length > 0
17                 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
18                 // PERMISO CONCEDIDO
19             } else {
20                 // PERMISO DENEGADO
21                 Toast.makeText(this,"É NECESARIO O PERMISO PARA CHAMAR POR TELÉFONO",Toast.LENGTH_LONG).show();
22             }
23             return;
24         }
25
26         // Comprobamos os outros permisos
27
28     }
29 }

```

- 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:

```

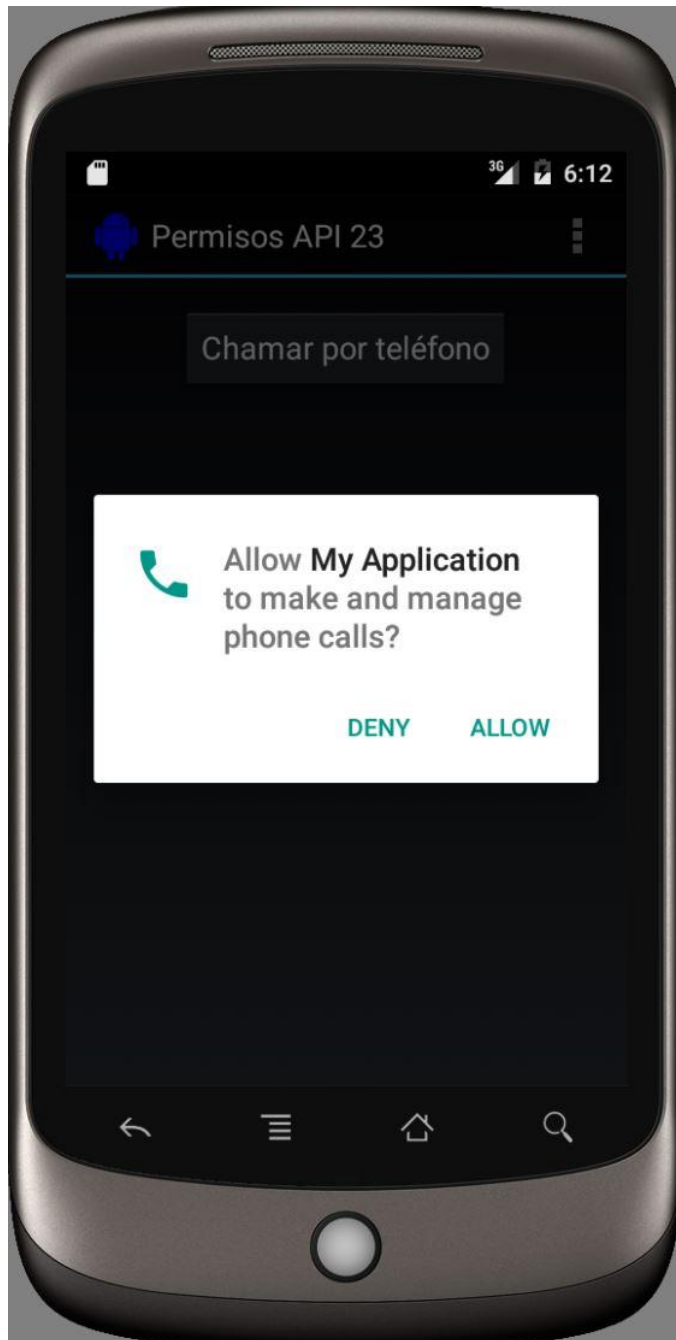
1
2 public void pedirPermiso(){
3
4     ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.CALL_PHONE},CODIGO_IDENTIFICADOR);
5
6 }

```

## 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.



## Creamos a activity

- Nome do proxecto: **UD08\_01\_Permisos**
- Nome da activity: **UD08\_01\_Permisos.java**

## Código do layout xml

```
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
3   android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
4   android:paddingRight="@dimen/activity_horizontal_margin"
5   android:paddingTop="@dimen/activity_vertical_margin"
6   android:paddingBottom="@dimen/activity_vertical_margin"
7   tools:context="es.cifprodolfoucha.angelfernandez.myapplication.UD08_01_Permisos">
8
9   <Button
10     android:layout_width="wrap_content"
11     android:layout_height="wrap_content"
12     android:text="Chamar por teléfono"
```

```

13     android:id="@+id/UD08_01_btnChamar"
14     android:layout_alignParentTop="true"
15     android:layout_centerHorizontal="true" />
16 </RelativeLayout>

```

### Código da clase UDo8\_01\_Permisos

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

**Nota:** Non se utilizan bibliotecas de compatibilidade.

```

1 public class UD08_01_Permisos extends Activity {
2
3     // Usado por si necesitamos diferentes permisos, para identificar cual de ellos es
4     private final int CODIGO_IDENTIFICADOR=1;
5
6     private void chamarTelefono(){
7         Intent callIntent = new Intent(Intent.ACTION_CALL);
8         callIntent.setData(Uri.parse("tel:123456789"));
9         startActivity(callIntent);
10    }
11
12    private void xestionarEventos(){
13
14        Button boton = (Button)findViewById(R.id.UD08_01_btnChamar);
15        boton.setOnClickListener(new View.OnClickListener() {
16            @Override
17            public void onClick(View view) {
18
19                if (Build.VERSION.SDK_INT==23){
20                    int permiso = checkSelfPermission(Manifest.permission.CALL_PHONE);
21                    if (permiso ==PackageManager.PERMISSION_GRANTED){
22                        chamarTelefono();
23                    }
24                    else{
25                        UD08_01_Permisos.this.requestPermissions( new String[]{Manifest.permission.CALL_PHONE},CODIGO_IDENTIFICADOR);
26                    }
27                }
28            }
29        });
30    }
31
32
33
34    @Override
35    protected void onCreate(Bundle savedInstanceState) {
36        super.onCreate(savedInstanceState);
37        setContentView(R.layout.activity_ud08_01_permisos);
38
39        xestionarEventos();
40    }
41
42
43
44    @Override
45    public void onRequestPermissionsResult(int requestCode,
46        String permissions[], int[] grantResults) {
47        switch (requestCode) {
48            case CODIGO_IDENTIFICADOR: {
49                // If request is cancelled, the result arrays are empty.
50                if (grantResults.length > 0
51                    && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
52                    chamarTelefono();
53                } else {
54
55                    Toast.makeText(this,"É NECESARIO O PERMISO PARA CHAMAR POR TELÉFONO",Toast.LENGTH_LONG).show();
56                }
57                return;
58            }
59
60            // other 'case' lines to check for other
61            // permissions this app might request
62        }
63    }
64
65    @Override
66    public boolean onCreateOptionsMenu(Menu menu) {
67        // Inflate the menu; this adds items to the action bar if it is present.
68        getMenuInflater().inflate(R.menu.menu_telefono_carrion, menu);
69        return true;
70    }
71

```

```
72  @Override
73  public boolean onOptionsItemSelected(MenuItem item) {
74      // Handle action bar item clicks here. The action bar will
75      // automatically handle clicks on the Home/Up button, so long
76      // as you specify a parent activity in AndroidManifest.xml.
77      int id = item.getItemId();
78
79      //noinspection SimplifiableIfStatement
80      if (id == R.id.action_settings) {
81          return true;
82      }
83
84      return super.onOptionsItemSelected(item);
85  }
86 }
```

-- Ángel D. Fernández González

---

Obtenido de «[https://manuais.iessanclemente.net/index.php?title=PDM\\_Avanzado\\_Permisos\\_AndroidManifest.xml&oldid=60707](https://manuais.iessanclemente.net/index.php?title=PDM_Avanzado_Permisos_AndroidManifest.xml&oldid=60707)»

---

**Esta página se editó por última vez el 24 oct 2016 a las 06:58.**

El contenido está disponible bajo la licencia [Creative Commons: CC-BY-NC-SA](#), a menos que se indique lo contrario.