Permisos (1)

Los permisos de Android establecen las operaciones que un determinado proceso o aplicación puede realizar, y son por tanto un mecanismo de seguridad.

- Cada aplicación se ejecuta en un entorno aislado y cuando necesita acceder a recursos de otra aplicación o del sistema, requiere un permiso.
- Al instalar una aplicación se mostrará una lista con los permisos que solicita para que el usuario sea consciente de los riesgos.

2

Permisos (2)

Tipos de permisos

- Normales: no afectan a la vida privada del usuario ni a sus datos personales (acceso a Internet, bluetooth, creación de iconos), de ahí que estén garantizados siempre que se soliciten en el manifiesto.
- Sensibles: afectan a la vida privada del usuario (almacenamiento, localización, teléfono, SMS, contactos, calendario, cámara, micrófono y sensor de ritmo cardíaco).

3

Permisos (3)

Tipos de permisos

La forma en que se conceden estos permisos depende de la versión de Android:

- Android 5.1 (API 22) o anterior: los permisos se solicitan en el momento de la instalación de la aplicación, y en caso de no autorizarse la aplicación no se instala.

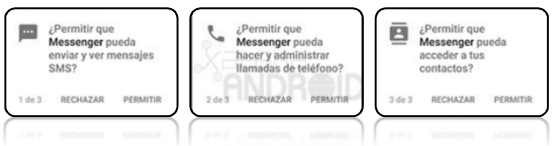
  Esto hace que la mayoría de los usuarios acabe por aceptar todos los permisos que se le solicitan.

4

Permisos (4)

Tipos de permisos

- Android 6.0 (API 23) o posterior: requieren una solicitud al usuario en tiempo de ejecución. Además se recomienda que la aplicación indique para qué necesita esos permisos de manera que el usuario tenga más elementos de juicio para conceder o no los permisos.



5

Permisos (5)

Solicitar un permiso: Permiso Normal

Para solicitar un permiso normal basta con agregar la etiqueta ***<uses-permission>*** en el archivo "AndroidManifest.xml".

Ejemplo: conexión a Internet

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

Los permisos de sistema empiezan por "*android.permission*", pueden consultarse en:

https://developer.android.com/reference/android/Manifest.permission.html

6

Permisos (6)

Solicitar un permiso: Permiso Sensible

Los permisos sensibles también tienen que declararse en el manifiesto, pero además en tiempo de ejecución tendremos que comprobar si disponemos de los permisos necesarios cada vez que ejecutemos algún proceso que necesite esos permisos.

7

Permisos (7)

Solicitar un permiso: Permiso Sensible

1) Comprobar si se ha asignado un permiso por parte del usuario. Para ello se utiliza el método *checkSelfPermission*, que devuelve un valor entero en forma de constante:
   - *PERMISSION_GRANTED*: permiso concedido.
   - *PERMISSION_DENIED*: permiso denegado.

8

Permisos (8)

Solicitar un permiso: Permiso Sensible

2) Si el usuario hubiera denegado el permiso podríamos ofrecerle una explicación acerca de la necesidad del permiso en la aplicación.

   Podemos comprobar si un permiso ya se ha solicitado mediante el método:

   shouldShowRequestPermissionRationale
   - *true*: solicitud realizada y rechazada.
   - *false*: existe una política del dispositivo que impide a la app este permiso, o el usuario ha indicado que no se le vuelva a preguntar.

9

Permisos (9)

Solicitar un permiso: Permiso Sensible

3) Solicitar el permiso mediante el método *requestPermission*, que recibirá como parámetros la actividad, un array de *string* con los permisos solicitados*, y un código de respuesta en forma de constante.

   * Esos permisos tienen que estar declarados en el manifiesto y ser de tipo sensible.

10

Permisos (10)

Solicitar un permiso: Permiso Sensible

4) Recuperar el resultado de un permiso. Para ello hemos de sobrecargar el método *onRequestPermissionsResult* que recibirá el código de repuesta asociado a la petición, el array de permisos y el array de resultados para cada permiso.

   Sólo podremos realizar la operación cuando el permiso haya sido concedido.

11

Permisos (11)

Solicitar un permiso: Permiso Sensible

```
// Comprobamos si no disponemos del permiso: READ_CONTACTS
if (ContextCompat.checkSelfPermission(thisActivity,
   Manifest.permission.READ_CONTACTS)!=PackageManager.PERMISSION_GRANTED) {

   // Comprobamos si ya se había solicitado el permiso
   if (ActivityCompat.shouldShowRequestPermissionRationale(thisActivity,
      Manifest.permission.READ_CONTACTS)) {
      // Mostraríamos información al usuario de la necesidad del permiso
      // y después se volvería a solicitar el permiso

   } else {
      // No es necesario mostrar información adicional
      ActivityCompat.requestPermissions(thisActivity,
         new String[]{Manifest.permission.READ_CONTACTS},
         REQUEST_READ_CONTACTS);
   }
}
```

12

Permisos (12)

Solicitar un permiso: Permiso Sensible

```
@Override
public void onRequestPermissionsResult (int requestCode,
String[] permissions, int[] grantResults ) {
  switch (requestCode) {
    case REQUEST_READ_CONTACTS:
      // Si la solicitud se anula el resultado (grantResult) estará vacío
      if (grantResults.length>0 &&
          grantResults[0]==PackageManager.PERMISSION_GRANTED) {
          // Permiso concedido, realizamos la operación
      } else {
          // Permiso denegado, no podemos realizar la operación
      }
      return;

    // Otros case con el resto de solicitudes (requestCode)
  }
}
```

13

Permisos (13)

Declarar un permiso

Podemos declarar nuestros propios permisos en el manifiesto mediante la etiqueta ***<permission>***, que requiere tres atributos:

- label: nombre corto que sirve para identificar el permiso.
- description: texto más completo que describe los datos que son accesibles.
- protectionLevel: *normal* o *dangerous*.

14

Permisos (14)

Declarar un permiso

Ejemplo:

```
<permission android:name="com.example.oscar.expermisos.DEADLY_ACTIVITY"
    android:label="@string/label_DEADLY_ACTIVITY"
    android:description="@string/description_DEADLY_ACTIVITY"
    android:protectionLevel="dangerous" />
```

Los valores de "*label*" y "*description*" suelen establecerse por medio de *strings* que se almacenan en la carpeta *values* de los recursos de la aplicación.

15