

CAPÍTULO 6. Seguridad y posicionamiento

6.1. Seguridad en Android

6.1.1.1. Creando tus propios permisos

Para definir el nuevo permiso utilizaremos el tag `<permission>` en el fichero `AndroidManifest.xml` de nuestro *software*. A continuación se muestra un ejemplo:

```
<manifest package="com.payperview.servicios">
    ...
    <permission android:name="com.payperview.servicios.VER_VIDEOS"
        android:label="@string/etiqueta"
        android:description="@string/descripcion"
        android:permissionGroup="android.permission-group.COST_MONEY"
        android:protectionLevel="dangerous" />
</manifest>
```

Los atributos `android:label` y `android:description` son opcionales y han de ser introducidos a través de un recurso de cadena. En estas cadenas hay que describir el permiso de forma abreviada y extensa, respectivamente. Veamos cómo podría ser en el ejemplo:

```
<string name="etiqueta">reproducción de videos bajo demanda</string>
<string name="descripcion">Permite a la aplicación reproducir videos de
    la empresa PayPerView sin tu intervención. Se trata de un servicio
    no gratuito, por lo que puede afectar al saldo de tu cuenta con
    esta empresa. Si no tienes una cuenta abierta los videos no podrán
    ser reproducidos. </string>
```

Vamos a comprobar cómo usar este permiso con el siguiente ejemplo:

Crea una nueva actividad en este proyecto con nombre `VerVideo` y copia el mismo código de la actividad `NuevoPermiso`. En el ejemplo real esta actividad sería la responsable de la reproducción de videos. Modifica el fichero `AndroidManifest.xml` según el código mostrado a continuación:

```
<manifest...>
    <application...>
        ...
        <activity
            android:name="VerVideo"
            android:permission="com.payperview.servicios.VER_VIDEOS">
```

```

        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="
                android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
</application>
</manifest>

```

Copia detrás de `</application>` la etiqueta `<permission .../>` del ejemplo anterior. Recuerda definir los recursos de cadena `etiqueta` y `descripcion`.

Ejecuta el proyecto. Es imprescindible para registrar en el teléfono el nuevo permiso y la nueva actividad que queremos lanzar desde otras aplicaciones. Para usar este servicio crea un nuevo proyecto con los siguientes datos:

```

Project name: UsarPermiso
Build Target: Android1.5
Application name: UsarPermiso
Package name: org.example.usarpermiso
Create Activity: UsarPermiso
Min SDK Version: 3

```

Abre el fichero `main.xml` e inserta el siguiente botón dentro del `<LinearLayout>`:

```

<Button android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ver Vídeo"
    android:id="@+id/Button01"/>

```

Abre el fichero `UsarPermisos.java` y añade el siguiente código al final de la función `onCreate()`:

```

Button b = (Button)findViewById(R.id.Button01);
b.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent i = new Intent();
        i.setClassName("com.payperview.servicios",
            "com.payperview.servicios.VerVideo");
        startActivity(i);
    }
});

```

Ejecuta la aplicación. Cuando pulses el botón la aplicación provocará un error. Visualiza la ventana *LogCat* para verificar que se trata de un error de permiso.

Para solucionar el problema tendrás que incluir el siguiente código al final del fichero `AndroidManifest.xml`:

```

<manifest ...>
    ...
    <uses-permission android:name="com.payperview.servicios.VER_VIDEOS"/>
</manifest>

```

Comprueba cómo ahora se accede al servicio sin problemas.

6.2. Localización

```
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:id="@+id/salida"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
</ScrollView>
```

Ve ahora al fichero `LocationTest.java` y sustituye el siguiente código:

```
public class Localizacion extends Activity implements LocationListener {

    private static final String[] A = { "n/d", "preciso", "impreciso" };
    private static final String[] P = { "n/d", "bajo", "medio", "alto" };
    private static final String[] E = { "fuera de servicio",
                                         "temporalmente no disponible ", "disponible" };

    private LocationManager manejador;
    private TextView salida;
    private String proveedor;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        salida = (TextView) findViewById(R.id.salida);

        manejador = (LocationManager) getSystemService(LOCATION_SERVICE);
        log("Proveedores de localización: \n ");
        muestraProveedores();

        Criteria criteria = new Criteria();
        proveedor = manejador.getBestProvider(criteria, true);
        log("Mejor proveedor: " + proveedor + "\n");

        log("Comenzamos con la última localización conocida:");
        Location localizacion = manejador.getLastKnownLocation(proveedor);
        muestraLocaliz(localizacion);
    }

    // Métodos del ciclo de vida de la aplicación
    @Override
    protected void onResume() {
        super.onResume();
        // Activamos notificaciones de localización
        manejador.requestLocationUpdates(proveedor, 10000, 1, this);
    }

    @Override
    protected void onPause() {
        super.onPause();
        // Desactivamos notificaciones para ahorrar batería
        manejador.removeUpdates(this);
    }
}
```

```

// Métodos de la interfaz LocationListener
public void onLocationChanged(Location location) {
    log("Nueva localización:");
    muestraLocaliz(location);
}

public void onProviderDisabled(String proveedor) {
    log("Proveedor deshabilitado: " + proveedor + "\n");
}

public void onProviderEnabled(String proveedor) {
    log("Proveedor habilitado:" + proveedor + "\n");
}

public void onStatusChanged(String proveedor, int estado,
    Bundle extras) {
    log("Cambia estado proveedor: " + proveedor + ", estado="
        + E[Math.max(0,estado)] + ", extras=" + extras + "\n");
}

// Métodos para mostrar información
private void log(String cadena) {
    salida.append(cadena + "\n");
}

private void muestraLocaliz(Location location) {
    if (location == null)
        log("Localización desconocida\n");
    else
        log(location.toString() + "\n");
}

private void muestraProveedores() {
    List<String> proveedores = manejador.getAllProviders();
    for (String proveedor : proveedores) {
        muestraProveedor(proveedor);
    }
}

private void muestraProveedor(String proveedor) {
    LocationProvider info = manejador.getProvider(proveedor);
    log("LocationProvider[" + "getName=" + info.getName() +
        ", isProviderEnabled=" + manejador.isProviderEnabled(proveedor) +
        ", getAccuracy=" + A[Math.max(0, info.getAccuracy())] +
        ", getPowerRequirement=" +
        P[Math.max(0, info.getPowerRequirement())] +
        ", hasMonetaryCost=" + info.hasMonetaryCost() +
        ", requiresCell=" + info.requiresCell() +
        ", requiresNetwork=" + info.requiresNetwork() +
        ", requiresSatellite=" + info.requiresSatellite() +
        ", supportsAltitude=" + info.supportsAltitude() +
        ", supportsBearing=" + info.supportsBearing() +
        ", supportsSpeed=" + info.supportsSpeed() + " ]\n");
}
}

```

6.3. Google Maps

6.3.1. Un programa de ejemplo

El fichero `AndroidManifest.xml` se muestra a continuación:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.example.ejemplogooglemaps" android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".EjemploGoogleMaps"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <uses-library android:name="com.google.android.maps"/>
    </application>
    <uses-sdk android:minSdkVersion="3"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission
        android:name="android.permission.ACCESS_FINE_LOCATION"/>
</manifest>
```

Sustituye el `layout/main.xml` por el siguiente fichero:

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

    <com.google.android.maps.MapView
        android:id="@+id/mapa"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:clickable="true"
        android:apiKey="">>>> Tu clave de Google Maps <<<"/>
</RelativeLayout>
```

Sustituye `">>> Tu clave de Google Maps <<<` por la clave de Google Maps obtenida en el apartado anterior. Reemplaza el código de `EjemploGoogleMaps.java` por:

```

public class EjemploGoogleMaps extends MapActivity {
    private MapController mapController;
    private MapView mapView;
    private LocationManager locationManager;

    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        setContentView(R.layout.main);
        mapView = (MapView) findViewById(R.id.mapa);
        mapView.setBuiltInZoomControls(true); //Activa controles zoom
        mapView.setSatellite(true); //Activa vista satélite
        mapView.setStreetView(false); //Desactiva StreetView
        mapView.setTraffic(false); //Desactiva información de tráfico
        mapController = mapView.getController();
        mapController.setZoom(14); // Zoon 1 ver todo el mundo
        locationManager = (LocationManager)

        getSystemService(Context.LOCATION_SERVICE);
        locationManager.requestLocationUpdates(LocationManager.
                                                    GPS_PROVIDER, 0, 0, new
        GeoUpdateHandler());
    }

    @Override
    protected boolean isRouteDisplayed() {
        return false;
    }

    public class GeoUpdateHandler implements LocationListener {

        @Override public void onLocationChanged(Location location) {
            int lat = (int) (location.getLatitude() * 1E6);
            int lng = (int) (location.getLongitude() * 1E6);
            GeoPoint point = new GeoPoint(lat, lng);
            mapController.setCenter(point);
        }

        @Override public void onProviderDisabled(String provider) {}

        @Override public void onProviderEnabled(String provider) {}

        @Override
        public void onStatusChanged(String provider, int status,
            Bundle extras){}
    }
}

```

6.4. Fragmentando los asteroides

Convierte la variable local `drawableAsteroide` del constructor, en una variable global, que será un *array* de tres elementos:

```
private Drawable drawableAsteroide[] = new Drawable[3];
```

En el constructor cuando se quiera trabajar con bitmaps inicializaremos esta variable de la siguiente forma:

```
drawableAsteroide[0] = context.getResources().getDrawable(R.drawable.asteroide1);
drawableAsteroide[1] = context.getResources().getDrawable(R.drawable.asteroide2);
drawableAsteroide[2] = context.getResources().getDrawable(R.drawable.asteroide3);
```

Inicializa también las variables en caso de querer trabajar con gráficos vectoriales.

Añade al principio del método `destruyeAsteroide(int i)` el siguiente código:

```
int tam;
if(Asteroides.get(i).getDrawable()!=drawableAsteroide[2]){
    if(Asteroides.get(i).getDrawable()==drawableAsteroide[1]){
        tam=2;
    } else {
        tam=1;
    }
}
for(int n=0;n<numFragmentos;n++){
    Grafico asteroide = new Grafico(this,drawableAsteroide[tam]);
    Asteroid.setPosX(Asteroides.get(i).getPosX());
    asteroide.setPosY(Asteroides.get(i).getPosY());
    asteroide.setIncX(Math.random()*7-2-tam);
    asteroide.setIncY(Math.random()*7-2-tam);
    asteroide.setAngulo((int)(Math.random()*360));
    asteroide.setRotacion((int)(Math.random()*8-4));
    Asteroides.add(asteroide);
}
}
```

Corrige algún error adicional ocasionado por este cambio.