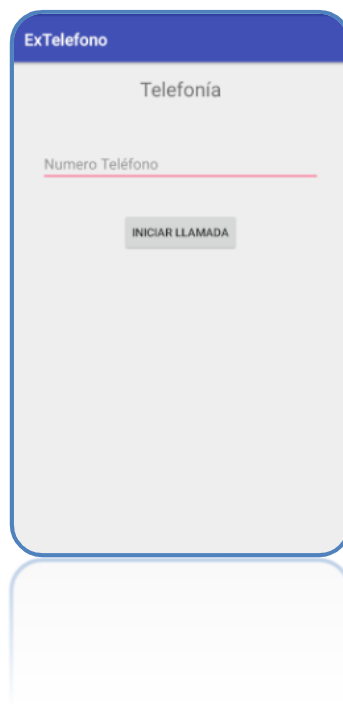


## UNIDAD 4: Llamadas de Teléfono

Vamos a ver como se gestionan las llamadas de teléfono desde nuestra aplicación, tanto para realizar llamadas como para recibir información de las llamadas entrantes y poder reaccionar ante ellas.

En cuanto a las llamadas entrantes, en este ejemplo simplemente mostraremos una alerta que indique el cambio de estado del teléfono (sonando, comunicando, inactivo), y además en caso de pasar a inactivo tras una llamada reiniciaremos nuestra aplicación.



## SOLICITUD DE PERMISOS

Nuestra aplicación requiere el uso del hardware del teléfono para poder realizar llamadas, así que lo indicaremos en nuestro archivo de "Manifiesto" para que *Google Play* no ofrezca la aplicación a los dispositivos que no dispongan de telefonía:

```
<uses-feature android:name="android.hardware.telephony" android:required="true" />
```

Además, necesitará acceder al estado del teléfono, así que también tendremos que solicitar el permiso `READ_PHONE_STATE`:

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

El archivo de "Manifiesto" sería similar al siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.oscar.extelefono">

    <uses-feature android:name="android.hardware.telephony" android:required="true"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

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

    </application>
</manifest>
```

## CREACIÓN DE LA INTERFAZ

La interfaz contendrá una única pantalla en la que añadiremos un título, una caja de texto (*editTelefono*) para que el usuario introduzca el número de teléfono al que llamar (tipo *phone*) y un botón (*botTelefono*) para iniciar la llamada.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.example.oscar.extelefono.MainActivity"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/txtTitulo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Telefonía"
        android:textSize="24sp"
        android:gravity="center"
        android:layout_margin="16dp"/>

    <EditText
        android:id="@+id/editTelefono"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="phone"
        android:hint="Numero Teléfono"
        android:layout_margin="36dp" />

    <Button
        android:id="@+id/botTelefono"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Iniciar llamada"
        android:layout_gravity="center"/>

</LinearLayout>
```

## CLASE MYPHONELISTENER

Para gestionar las llamadas entrantes vamos a implementar una clase interna (*MyPhoneListener*) que extiende a *PhoneStateListener* dentro de nuestro *MainActivity*.

```
private class MyPhoneListener extends PhoneStateListener {  
    ...  
}
```

En esta clase definiremos un único atributo *booleano onCall* que guardará en cada momento el estado de la llamada: activa (*true*) o inactiva (*false*).

```
private boolean onCall = false;
```

Y sobrecargaremos el método *onCallStateChanged* que se encargará de gestionar cada uno de los estados del teléfono cuando se produzca un cambio en el estado, para ello recibirá dos parámetros:

- **state**: entero que contiene el nuevo estado del teléfono en forma de constante: recibiendo llamada (*CALL\_STATE\_RINGING*), en llamada o comunicando (*CALL\_STATE\_OFFHOOK*) o inactivo (*CALL\_STATE\_IDLE*).
- **incomingNumber**: *String* con el número de teléfono de la llamada recibida.

```
@Override  
public void onCallStateChanged(int state, String incomingNumber) {  
    String mensa = "";  
    int duracion = Toast.LENGTH_LONG;  
    Toast toast;  
    switch (state) {  
        ...  
    }  
}
```

Para cada uno de los estados realizaremos las siguientes acciones:

- **CALL\_STATE\_RINGING**: mostraremos una alerta indicando la recepción de una llamada.

```
case TelephonyManager.CALL_STATE_RINGING:
    // Sonando
    mensa = "Llamada Sonando";
    toast = Toast.makeText(MainActivity.this, mensa, duracion);
    toast.show();
    break;
```

- **CALL\_STATE\_OFFHOOK**: mostraremos una alerta indicando que estamos en comunicación y cambiaremos el estado de la variable *onCall* a *true*.

```
case TelephonyManager.CALL_STATE_OFFHOOK:
    // En llamada/Comunicando
    mensa = "Llamada Comunicando";
    toast = Toast.makeText(MainActivity.this, mensa, duracion);
    toast.show();
    onCall = true;
    break;
```

- **CALL\_STATE\_IDLE**: si el estado anterior era llamada (*onCall* = *true*), mostraremos una alerta, cambiaremos el valor de *onCall* a *false* y devolveremos el control a nuestra aplicación reiniciando la actividad.

```
case TelephonyManager.CALL_STATE_IDLE:
    // Inactivo
    if (onCall == true) {
        // Inactivo tras llamada
        mensa = "Reinicio de aplicacion tras llamada";
        toast = Toast.makeText(MainActivity.this, mensa, duracion);
        toast.show();
        onCall = false;
        // Reiniciamos la aplicación
        Intent restart = getBaseContext().getPackageManager().
            getLaunchIntentForPackage(getBaseContext().getPackageName());
        restart.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(restart);
    }
    break;
```

La clase completa quedaría así:

```
private class MyPhoneListener extends PhoneStateListener
{
    private boolean onCall = false;

    @Override
    public void onCallStateChanged(int state, String incomingNumber)
    {
        String mensa = "";
        int duracion = Toast.LENGTH_LONG;
        Toast toast;
        switch (state) {

            case TelephonyManager.CALL_STATE_RINGING:
                // Sonando
                mensa = "Llamada Sonando";
                toast = Toast.makeText(MainActivity.this, mensa, duracion);
                toast.show();
                break;

            case TelephonyManager.CALL_STATE_OFFHOOK:
                // Descolgado/Comunicando
                mensa = "Llamada Comunicando";
                toast = Toast.makeText(MainActivity.this, mensa, duracion);
                toast.show();
                onCall = true;
                break;

            case TelephonyManager.CALL_STATE_IDLE:
                // Inactivo
                if (onCall == true) {
                    // Inactivo tras llamada
                    mensa = "Reinicio de aplicacion tras llamada";
                    toast = Toast.makeText(MainActivity.this, mensa, duracion);
                    toast.show();
                    onCall = false;

                    // Reiniciamos la aplicación
                    Intent restart = getBaseContext().getPackageManager().
                        getLaunchIntentForPackage(getBaseContext().getPackageName());
                    restart.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(restart);
                }
                break;

            default:
                break;
        }
    }
}
```

## CLASE MAINACTIVITY

En la clase *MainActivity* además de establecer la vista gestionaremos tanto las llamadas entrantes como la realización de llamadas.

- **Gestionar las llamadas entrantes:** crearemos un objeto de nuestra clase *MyPhoneListener*, y *TelephonyManager* al que añadiremos el método *listen* para que escuche los cambios de estado de nuestro objeto.

```
MyPhoneListener phoneListener = new MyPhoneListener();
TelephonyManager tm =
    (TelephonyManager) this.getSystemService(Context.TELEPHONY_SERVICE);
tm.listen(phoneListener, PhoneStateListener.LISTEN_CALL_STATE);
```

- **Realizar llamadas:** obtendremos una referencia al botón *botTelefono*, le añadiremos un *listener* e implementaremos el método *onClick*.

Dentro del método crearemos un objeto *PackageManager* para comprobar si disponemos de la función de telefonía (*FEATURE\_TELEPHONY*) utilizando el método *hasSystemFeature*.

En caso de disponer de telefonía, recogeremos el contenido de la caja de texto *EditTelefono* y realizaremos la llamada mediante un *Intent* cuyo action será *Intent.ACTION\_DIAL* y los datos la *URI* formada por "tel:" y el número de teléfono recogido.

Si no disponemos de telefonía o la caja de texto está vacía informaremos mediante un *Toast*.

```
Button botTelefono = (Button)findViewById(R.id.botTelefono);
botTelefono.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        PackageManager pm = getPackageManager();
        if (pm.hasSystemFeature(pm.FEATURE_TELEPHONY)) {
            EditText eTlf = (EditText)findViewById(R.id.editTelefono);
            String numTel = eTlf.getText().toString().trim();
            if (!numTel.equals("")) {
                Uri tlf = Uri.parse("tel:" + numTel);
                Intent call = new Intent(Intent.ACTION_DIAL, tlf);
                startActivity(call);
            } else {
                int duracion = Toast.LENGTH_SHORT;
                String mensa = "Debe introducir el número de teléfono";
                Toast t = Toast.makeText(MainActivity.this, mensa, duracion);
                t.show();
            }
        } else {
            int duracion = Toast.LENGTH_SHORT;
            String mensa = "Su dispositivo no puede realizar llamadas";
            Toast t = Toast.makeText(getApplicationContext(), mensa, duracion);
            t.show();
        }
    }
});
```

La clase *MainActivity* quedaría así:

```
public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Gestión de llamadas entrantes
        MyPhoneListener phoneListener = new MyPhoneListener();
        TelephonyManager tm =
            (TelephonyManager) this.getSystemService(Context.TELEPHONY_SERVICE);
        tm.listen(phoneListener, PhoneStateListener.LISTEN_CALL_STATE);

        // Realizar llamada
        Button botTelefono = (Button)findViewById(R.id.botTelefono);
        botTelefono.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                PackageManager pm = getPackageManager();
                if (pm.hasSystemFeature(pm.FEATURE_TELEPHONY)) {
                    EditText eTlf = (EditText)findViewById(R.id.editTelefono);
                    String numTel = eTlf.getText().toString().trim();
                    if (!numTel.equals("")) {
                        Uri tlf = Uri.parse("tel:" + numTel);
                        Intent call = new Intent(Intent.ACTION_DIAL, tlf);
                        startActivity(call);
                    } else {
                        int duracion = Toast.LENGTH_SHORT;
                        String mensa = "Debe introducir el número de teléfono";
                        Toast t = Toast.makeText(MainActivity.this, mensa, duracion);
                        t.show();
                    }
                } else {
                    int duracion = Toast.LENGTH_SHORT;
                    String mensa = "Su dispositivo no puede realizar llamadas";
                    Toast t = Toast.makeText(getApplicationContext(), mensa, duracion);
                    t.show();
                }
            }
        });
    }

    private class MyPhoneListener extends PhoneStateListener {
        ...
    }
}
```



## PRUEBAS

Si probamos la aplicación en un dispositivo virtual, al intentar realizar una llamada se comprobará que no disponemos de hardware de teléfono y no podremos realizar la llamada. Tampoco podremos comprobar la gestión de las llamadas entrantes, ya que no hay manera de llamar a nuestro dispositivo virtual para que se produzca un cambio de estado, y éste siempre será inactivo.

Para poder probar la aplicación tendremos que instalarla en un dispositivo físico (con tarjeta SIM), y aquí sí que podremos tanto realizar llamadas como ver qué ocurre cuando recibimos una llamada, descolgamos el teléfono y lo colgamos tras finalizar la llamada.