

# 1. Archivo AndroidManifest.xml: Permiso

Primero, debes declarar el permiso de llamada telefónica.

XML

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

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

    <application
        ...
        android:theme="@style/Theme.TuAplicacion">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

## 2. 🎨 Layout: activity\_main.xml

Este diseño contiene un simple botón que disparará la lógica de la llamada.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tv_info"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Presiona para llamar al +584142886735"
        android:textSize="18sp"
        android:textStyle="bold"
        android:padding="16dp"
        app:layout_constraintBottom_toTopOf="@+id/btn_call"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_chainStyle="packed"/>
```

```
<Button
    android:id="@+id/btn_call"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Llamar Ahora"
    android:layout_marginTop="32dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tv_info"/>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

---

### 3. Lógica Java: MainActivity.java

Esta es la clase principal que maneja la **lógica condicional** (la comprobación del permiso), la **solicitud** del permiso y, finalmente, la **ejecución** de la llamada.

Java

```
package com.example.tuaplicacion;
```

```

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private static final int PERMISSION_REQUEST_CALL_PHONE = 1;
    // El número destino
    private final String NUMERO_DESTINO = "+584142886735";

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

        Button callButton = findViewById(R.id.btn_call);

        callButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // La condición previa es el clic en el botón.

```

```
        // Dispara la verificación de permisos.  
        checkAndRequestCallPermission();  
    }  
});  
}
```

```
/**  
 * Verifica si el permiso CALL_PHONE está concedido y, si no lo está, lo solicita.  
 */  
private void checkAndRequestCallPermission() {  
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.CALL_PHONE)  
        != PackageManager.PERMISSION_GRANTED) {
```

```
        // Permiso NO concedido: Solicitarlo.  
        ActivityCompat.requestPermissions(this,  
            new String[]{Manifest.permission.CALL_PHONE},  
            PERMISSION_REQUEST_CALL_PHONE);
```

```
    } else {  
        // Permiso CONCEDIDO: Proceder directamente a la llamada.  
        makePhoneCall();  
    }  
}
```

```
/**  
 * Maneja la respuesta del usuario a la solicitud de permisos.  
 */  
@Override  
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
```

```

    if (requestCode == PERMISSION_REQUEST_CALL_PHONE) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            // El usuario concedió el permiso.
            makePhoneCall();
        } else {
            // El usuario denegó el permiso.
            Toast.makeText(this, "Permiso de llamada denegado. No se puede realizar la acción.", Toast.LENGTH_LONG).show();
        }
    }
}

/**
 * Ejecuta la llamada telefónica usando un Intent.ACTION_CALL.
 */
private void makePhoneCall() {
    // Asegúrate de usar el esquema 'tel:' para números de teléfono
    String dialUri = "tel:" + NUMERO_DESTINO;
    Intent callIntent = new Intent(Intent.ACTION_CALL);
    callIntent.setData(Uri.parse(dialUri));

    try {
        // Dado que ya comprobamos el permiso, esta llamada debería ser segura.
        startActivity(callIntent);
    } catch (SecurityException e) {
        // Esto solo debería ocurrir si la comprobación previa falló o el SO lo bloqueó.
        Toast.makeText(this, "Error al iniciar la llamada: Faltan permisos.", Toast.LENGTH_LONG).show();
        e.printStackTrace();
    }
}
}

```

## Resumen del Flujo:

1. El usuario presiona el botón "**Llamar Ahora**".
2. El método `checkAndRequestCallPermission()` evalúa la **condición** (la posesión del permiso `CALL_PHONE`).
3. Si el permiso ya existe, se llama directamente a `makePhoneCall()`.
4. Si el permiso **no existe**, se solicita al usuario.
5. La respuesta del usuario se maneja en `onRequestPermissionsResult()`.
6. Si el usuario otorga el permiso, se reintenta y ejecuta `makePhoneCall()`, iniciando la llamada al +584142886735.