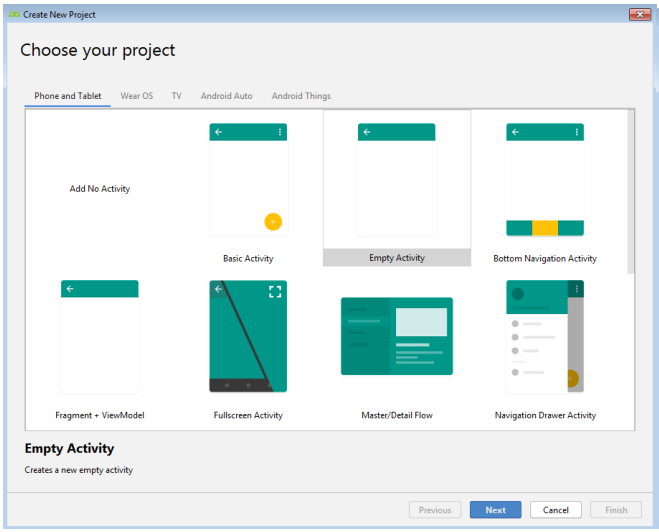
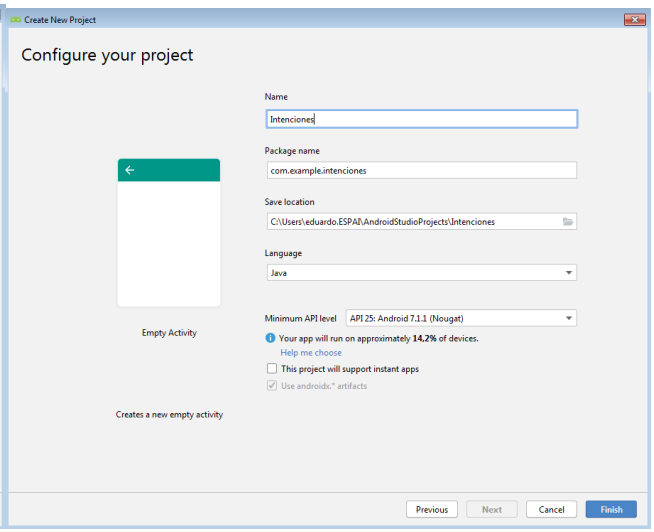
PRACTICA 9: INTENCIONES

Parte I: *Uso de intenciones implícitas*

Paso 1. Crea un nuevo proyecto con nombre Intenciones y tipo Empty Activity.





Paso 2. El *Layout* de la actividad inicial ha de estar formado por cinco botones, tal y como se muestra a continuación:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/btn1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="abrir página web"/>
    <Button
        android:id="@+id/btn2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="llamada de teléfono"/>
    <Button
        android:id="@+id/btn3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Google maps"/>
    <Button
        android:id="@+id/btn4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="tomar foto"/>
    <Button
        android:id="@+id/btn5"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="mandar correo"/>
</LinearLayout>
```

Paso 3. Abre la actividad principal e incorpora los siguientes métodos:

```java
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void pgWeb(View v) {
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.androidcurso.com/"));
        startActivity(intent);
    }
    public void llamadaTelefono(View view) {
        Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:962849347"));
        startActivity(intent);
    }
    public void googleMaps(View view) {
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("geo:41.656313,-0.877351"));
        startActivity(intent);
    }
    public void tomarFoto(View view) {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivity(intent);
    }
    public void mandarCorreo(View v) {
        Intent i = new Intent(Intent.ACTION_SEND);
        i.setType("text/plain");
        i.putExtra(Intent.EXTRA_SUBJECT, value: "asunto");
        i.putExtra(Intent.EXTRA_TEXT, value: "texto correo");
        i.putExtra(Intent.EXTRA_EMAIL, new String[]{"jtomas@upv.es"});
        startActivity(i);
    }
}
```

Paso 4. Asocia el atributo onClick de cada uno de los botones al método correspondiente.

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/btn1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="ABRIR PAGINA WEB"
        android:onClick="pgWeb"/>
    <Button
        android:id="@+id/btn2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="LLAMADA DE TELEFONO"
        android:onClick="llamadaTelefono"/>
    <Button
        android:id="@+id/btn3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="GOOGLE MAPS"
        android:onClick="googleMaps"/>
    <Button
        android:id="@+id/btn4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TOMAR FOTO"
        android:onClick="tomarFoto"/>
    <Button
        android:id="@+id/btn5"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="MANDAR CORREO"
        android:onClick="mandarCorreo"/>
</LinearLayout>
```

NOTA: *Si ejecutas esta aplicación en un emulador, es muy posible que los botones de mandar correo o GoogleMaps no funcionen. La razón es que no hay ninguna aplicación instalada en el emulador que sea capaz de realizar este tipo de acciones. Si tienes estos problemas, abre el* AVD Manager *y crea un dispositivo virtual con Google API. Estos dispositivos incorporan, además de las API de Android, algunas de las API de Google, como la de Google Maps (estas API se estudiarán más adelante).*

Paso 5. Abre AndroidManifest.xml e inserta la siguiente línea al final, antes de</manifest>:

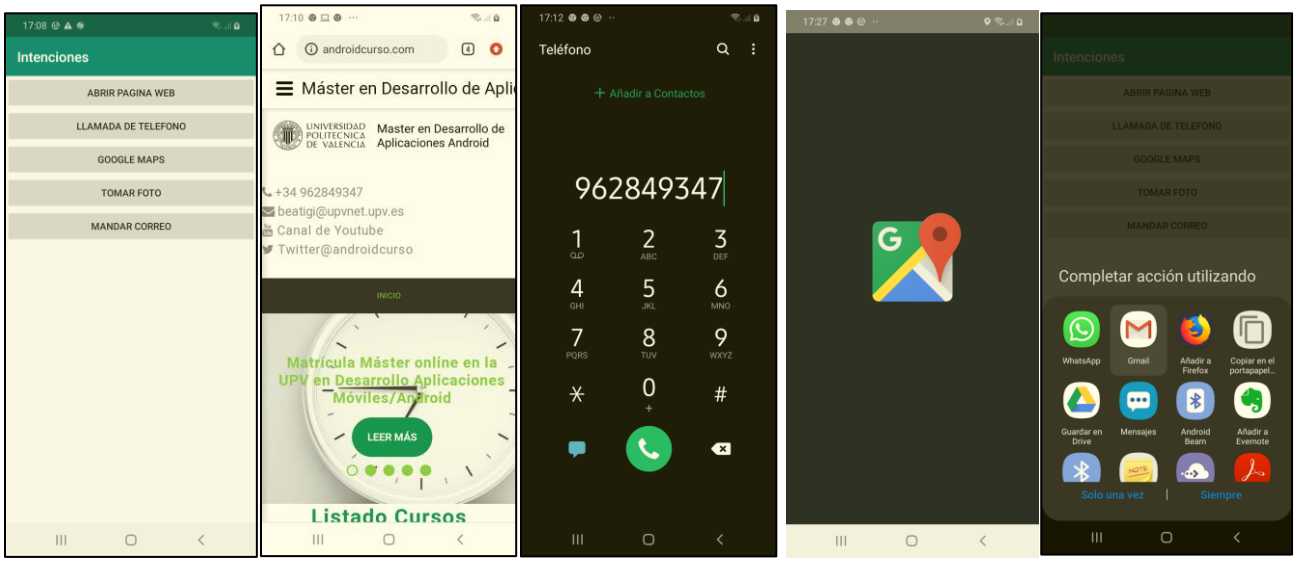
```xml
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.INTERNET"/>

</manifest>
```

NOTA: *En el CAPÍTULO 7 se estudiará el tema de la seguridad. Aprenderás como has de solicitar el permiso adecuado si quieres que tu aplicación llame por teléfono o acceda a Internet. Cuando estas acciones no las*

realizas directamente, si no que las pides a través de una intención, no es tu aplicación quien las realiza y por tanto no has de pedir estos permisos. La única excepción es el caso de realizar una llamada de teléfono. Para poder realizar una llamada de teléfono desde una intención si que hay que pedir el permiso correspondiente.

Paso 7. Ejecuta la aplicación en un terminal real. Observa todas las funcionalidades y los posibles errores que pueden ocurrir. Observa como el botón *mandar Correo* te permite seleccionar entre diferentes aplicaciones con esta funcionalidad



Tomar foto hace que la aplicación lance una excepción de seguridad:

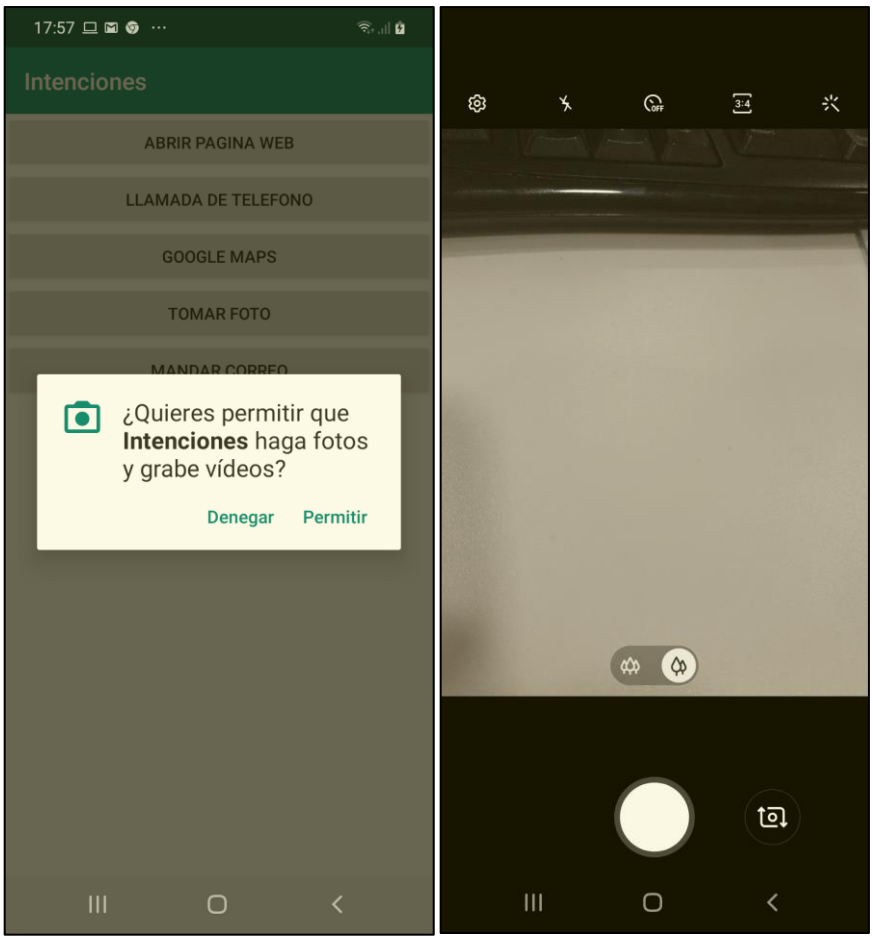
```
Caused by: java.lang.reflect.InvocationTargetException <1 internal call>
    at androidx.appcompat.app.AppCompatViewInflater$DeclaredOnClickListener.onClick(AppCompatViewInflater.java:397) <9 more...> <1 internal call> <2 more...>
Caused by: java.lang.SecurityException: Permission Denial: starting Intent { act=android.media.action.IMAGE_CAPTURE cmp=com.sec.android.app.camera/.Camera }
    at android.os.Parcel.createException(Parcel.java:1966)
    at android.os.Parcel.readException(Parcel.java:1934)
```

Paso 8. Para evitar la excepción de seguridad en el action de tomar una foto:

```
public void tomarFoto(View view) {
    //Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    //Intent intent = new Intent("android.media.action.IMAGE_CAPTURE");
    //startActivity(intent);
    checkPermission_CAMERA();
}
void checkPermission_CAMERA() {
    if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.CAMERA)!= PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.shouldShowRequestPermissionRationale( activity: this, Manifest.permission.CAMERA)) {
        } else {
            ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.CAMERA}, requestCode: 24);
        }
    } else {
        callCamara();
    }
}
public void callCamara(){
    //Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    Intent intent = new Intent( action: "android.media.action.IMAGE_CAPTURE");
    startActivity(intent);
}
@Override
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
    switch (requestCode) {
        case 24:
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                // permission was granted, yay! Do the phone call
                callCamara();

            } else {
                // permission denied, boo! Disable the
                // functionality that depends on this permission.
            }
            break;
    }
}
```

Paso 8. Ejecutamos la aplicación y vemos el resultado:



Práctica 5 (Parte II): *Uso de intenciones implícitas*

Paso 1. Crea nuevos botones en la aplicación del ejercicio anterior y experimenta con otro tipo de acciones y URL. Puedes consultar la tabla anterior. A continuación tienes algunas propuestas:

Paso 2. Compara las acciones VIEW y WEB_SEARCH. ¿Encuentras alguna diferencia?

Paso 3. Compara las acciones CALL y DIAL. ¿Encuentras alguna diferencia?

```
public void llamadaTelefono(View view) {
    //Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:962849347"));
    Intent intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:962849347"));
    startActivity(intent);
}
```

licitly check to see if permission is available (with checkPermission) or explicitly handle a potential SecurityException mo

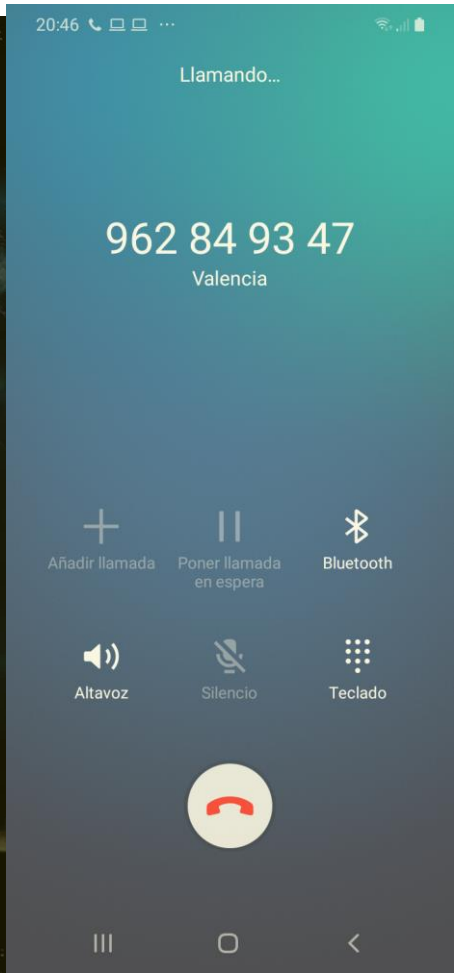
El sistema nos indica que se va a producir una excepción de seguridad y que se debe de utilizar la función checkPermission.

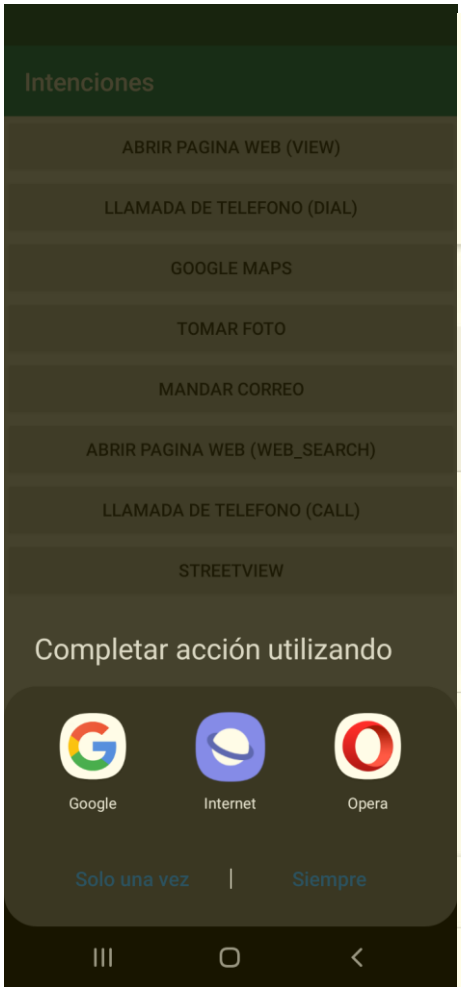
```
public void llamadaTelefono(View view) {
    //Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:962849347"));
    //Intent intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:962849347"));
    //startActivity(intent);
    checkPermission_CALL_PHONE();
}
void checkPermission_CALL_PHONE() {
    if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.CALL_PHONE)!= PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.shouldShowRequestPermissionRationale( activity: this, Manifest.permission.CALL_PHONE)) {
        } else {
            ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.CALL_PHONE},  requestCode: 42);
        }
    } else {
        callPhone();
    }
}
void callPhone() throws SecurityException{
    Intent i = new Intent(Intent.ACTION_CALL, Uri.parse("tel:" + "962849347"));
    startActivity(i);
}
@Override
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
    switch (requestCode) {
        case 24:
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                // permission was granted, yay! Do the phone call
                callCamara();

            } else {
                // permission denied, boo! Disable the
                // functionality that depends on this permission.
            }
            break;
        case 42:
            // If request is cancelled, the result arrays are empty.
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                // permission was granted, yay! Do the phone call
                callPhone();

            } else {
                // permission denied, boo! Disable the
                // functionality that depends on this permission.
            }
            break;
    }
}
```


20:45
¿Quieres permitir que **Intenciones** haga y gestione llamadas?
Denegar
Permitir


20:46
Llamando...
962 84 93 47
Valencia
Añadir llamada
Poner llamada en espera
Bluetooth
Altavoz
Silencio
Teclado


Intenciones
ABRIR PAGINA WEB (VIEW)
LLAMADA DE TELEFONO (DIAL)
GOOGLE MAPS
TOMAR FOTO
MANDAR CORREO
ABRIR PAGINA WEB (WEB_SEARCH)
LLAMADA DE TELEFONO (CALL)
STREETVIEW
Completar acción utilizando
Google
Internet
Opera
Solo una vez
Siempre


21:20
Google
guerra mundial
TODO
NOTICIAS
VÍDEOS
IMÁGENES
Segunda Guerr...
Conflicto militar
Guerra mundial
Guerra mundial es...
Primera Guerra Mundial
Seguir
28 jul. 1914 – 11 nov. 1918
RESUMEN
COMBATIENTES
VÍDEOS
La Primera Guerra Mundial, anteriormente llamada la Gran Guerra, fue una confrontación bélica centrada en Europa que empezó el 28 de julio de 1914 y finalizó el 11 de noviembre de 1918, cuando Alemania aceptó las condiciones
Discover
Notificaci...
Buscar
Recientes
Más

Paso 4. Experimenta con Google Streetview