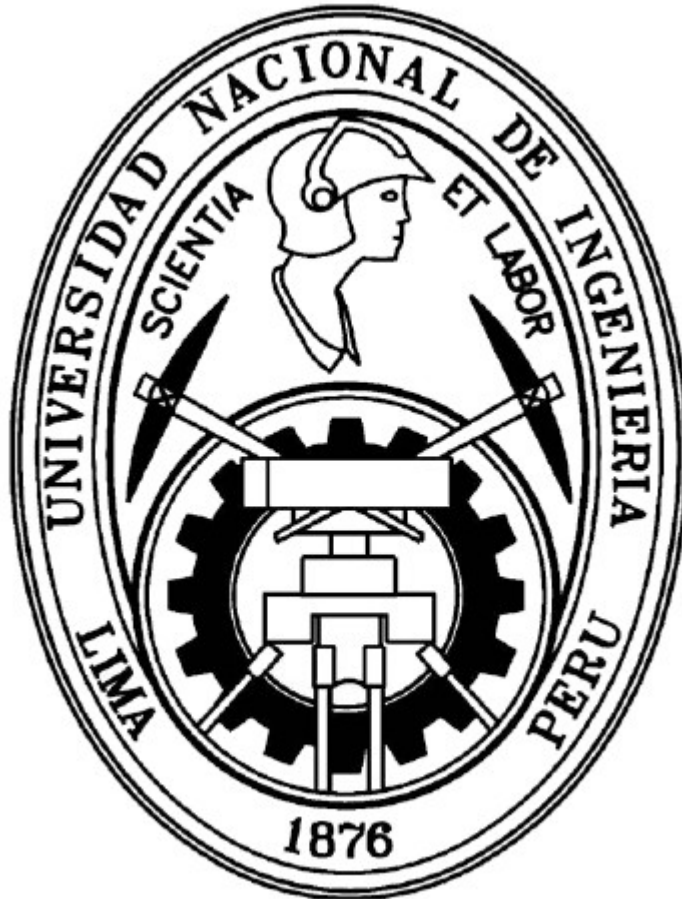


Laboratorio 4.2



Apellidos: Moreno Vera

Nombres: Felipe Adrian

Código: 20120354I

**Asignatura: Programación en Dispositivos Móviles
(CC481)**

2016 – I

Indice

Actividad 1 (3)

Actividad 2 (4)

Actividad 3 (5)

Actividad 4 (6)

Practica (8)

Actividad 1

1. Antes de empezar a programar nuestra Intent, lo primero que deberá realizar es crear un proyecto nuevo. Al estar creándolo en “Tittle” de la pantalla de Blank Activity le ponemos el nombre “Primera Pantalla”.
2. Abrimos el fichero AndroidManifest.xml y vemos que la actividad se encuentra como principal en la etiqueta <intent-filter>.
3. Quitamos el elemento TextView que aparecen en la vista diseño.
4. Creamos una nueva actividad (Blank Activity) dentro de nuestro paquete Java. A esta nueva clase le llamamos “Segunda”, el layout lo dejamos igual y el title le ponemos “Segunda Actividad”.
5. Recordemos que al crearla en AndroidManifest.xml se habrá actualizado con esta segunda actividad. También el nuevo fichero Java “Segunda” vemos que llama a esta segunda actividad.
6. Al igual que el laboratorio anterior en el diseño de la primera actividad insertamos un “Button” y modificamos las propiedades que consideremos oportunas, name, id recordemos de los recursos String.

El button

```
<Button
    android:layout_width="200dp"
    android:layout_height="50dp"
    android:text="Presioname :3 ><"
    android:id="@+id/btnPresiona"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:background="#e5fe03"
    android:textColor="#000000" />
```

El diseño



Actividad 2

1. Le damos la propiedad “salta” a nuestro evento onClick en la vista diseño.

Posteriormente, creamos el método correspondiente.

```
public void salta(View view){  
    // codigo omitido  
}
```

2. Un aspecto importante es utilizar el Log de Android. Esta clase nos sirve para hacer un seguimiento al código por consola. Hay muchos métodos como “e” de errores “v” de verbose, etc. En nuestro caso utilizaremos el método “d” que es el método de debug.

Para ello le pasamos dos parámetros App y el mensaje que queremos que muestre.

```
public void salta(View view){  
    Log.d(“App”,”se ha hecho click”);  
}
```

3. El siguiente paso es realizar un salto, una intención.

```
public void salta(View view){  
    Log.d(“App”,”se ha hecho click”);  
    Intent intent = new Intent(MyActivity.this, Segunda.class);  
    startActivity(intent);  
}
```

4. Recordemos que según el ciclo de vida que al pasar a la segunda actividad la primera todavía se encuentra en ejecución ya que no la hemos matado. Lo que vamos a realizar ahora es que en el segundo layout vamos a crear un Button

que nos devuelva a la primera actividad y finalice la segunda, es decir, que la mate. Para eliminar actividades completamente utilizamos el método finish().

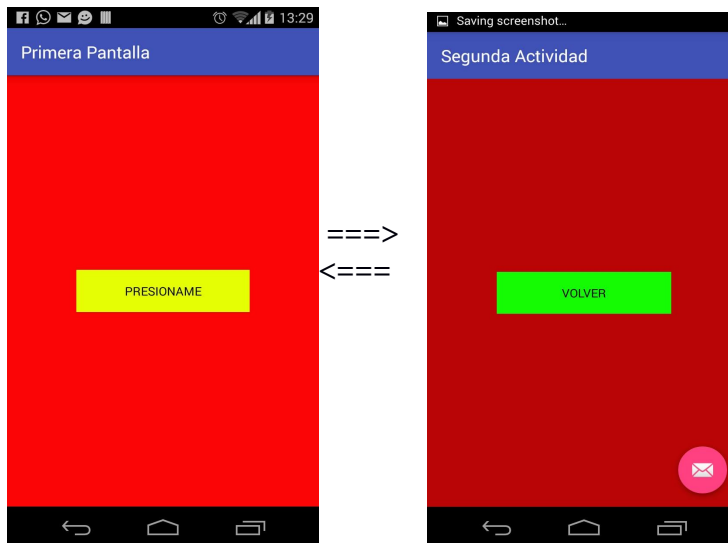
1. Creamos un Button en la segunda actividad con el valor “Volver”.

Todos los pasos vistos anteriormente.

2. Como anteriormente, hay que crear el método “Volver” en el código Java de nuestra segunda actividad. Recordad que el método finish() emula el botón volver del teléfono móvil.

```
public void volver(View view){  
    Log.d(“App”,”Adios con dios”)  
    finish();  
}
```

Ejecutando la app.



Actividad 3

1. Creamos una segunda clase Java que llamamos , Asegúrese de la herencia correcta.
2. En la segunda clase del código Java “SecondActivity” realizamos el método onCreate() si no se ha creado igual que el de la primera actividad.
3. En Actividad principal, en el método onCreate

1. Creamos nuestra Intent de la misma manera que venimos realizando.

2. Añadimos texto a nuestra Intent. Explique este método.

```
Intent.putExtra(“valorTest”,”true”);
```

añade un valor “true”, o cualquier cadena como mensaje, a la etiqueta “valorTest”.

3. Llamamos al método startActivity(Intent)

4. Declarar esta nueva actividad en AndroidManifest.xml

```
<activity android:name=".Tercera"
          android:label="@string/app_name2"/>
```

5. En nuestra segunda actividad en el método onCreate. Explicar el código que se expone a continuación:

```
String varString = getIntent().getStringExtra(“valorTest”);
```

```
Log.d(“HelloWorld – Second Acitivity”,varString);
```

getIntent, recibe un intent(esta a la espera de un intent),

getStringExtra. Recibe el mensaje “valorTest” como string y toda su info.

Y es guardado en varString.

6. Ejecutamos la aplicación y nos fijamos en la pantalla de debug. Comenta la salida.
Vemos que defrente va hacia la actividad a la que le dimos intent, sin mostrar la actividad principal.

```
nt D/HelloWorld,Second Acti: true
```

7. Cambiar el valor “true” por otro y fijese el cambio.

En primer lugar, la cadena HelloWorld Second Acitivity ... es demasiado largo (máximo 26 y tiene 29 caracteres).

```
nt D/HelloWorld,Second Acti: false
```

Probando con otra cadena ...

```
nt D/HelloWorld,Second Acti: Meow
```

Actividad 4

1. Creamos un nuevo proyecto.

2. Este proyecto vamos a hacer un Intent que cuando se pulse un Button nos direcciones a una nueva actividad con su XML

3. Este layout de la segunda actividad vamos a añadir 10 Buttons cada uno de ellos con una funcionalidad.

1. El primero que se utilizará será una redirección a una página web, para ello fijese de lo visto en teoría.

2. El segundo será para marcar un teléfono con ACTION_DIAL.

3. El tercero será una llamada directa a un teléfono con ACTION_CALL.

1. Se deberá añadir el siguiente permiso

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

4. El cuarto será para añadir contactos a nuestra agenda.

```
Intent intent= new Intent(Intent.ACTION_INSERT);  
intent.setType(ContactsContract.Contacts.CONTENT_TYPE);  
startActivity(intent);
```

5. El quinto crear o modificar una alarma.

```
Intent intent= new Intent(AlarmClock.ACTION_SET_ALARM);  
startActivity(intent);
```

En este punto 5, es necesario meterle permiso. `<uses-permission android:name="com.android.alarm.permission.SET_ALARM"/>`

6. El sexto es para enviar un correo electrónico. Explique este Intent.

```
Intent intent= new Intent(Intent.ACTION_SENDTO, Uri.fromParts(
"mailto", getResources().getString(R.string.mail), null));
intent.putExtra(Intent.EXTRA_SUBJECT, getResources().getString(
R.String.subject));
startActivity(Intent.createChooser(intent, getResources().getString(
R.string.envio)));
```

Aparece una opción para escoger el tipo de mail, relleno el subject(asunto) mediante el putExtra, destinatario (mailto), y el createChooser para escoger.

7. El séptimo permite iniciar la cámara fotográfica.

```
Intent intent= new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
startActivity(intent);
```

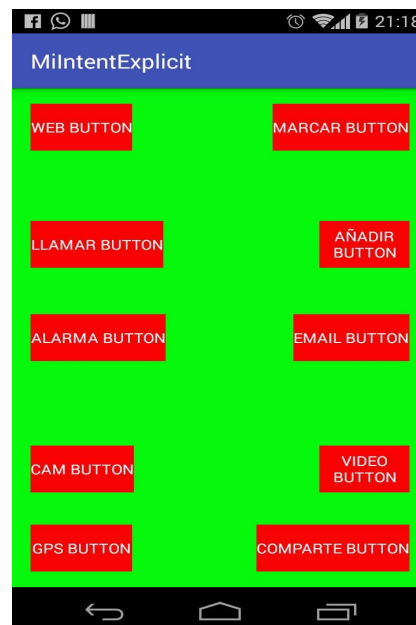
8. El octavo inicia la cámara de vídeo.

```
Intent intent=new Intent(MediaStore.INTENT_ACTION_VIDEO_CAMERA);
startActivity(intent);
```

9. El noveno mostrará las coordenadas GPS. Mirar Teoría.

10. El décimo comparte un texto seleccionando una aplicación del sistema Android.

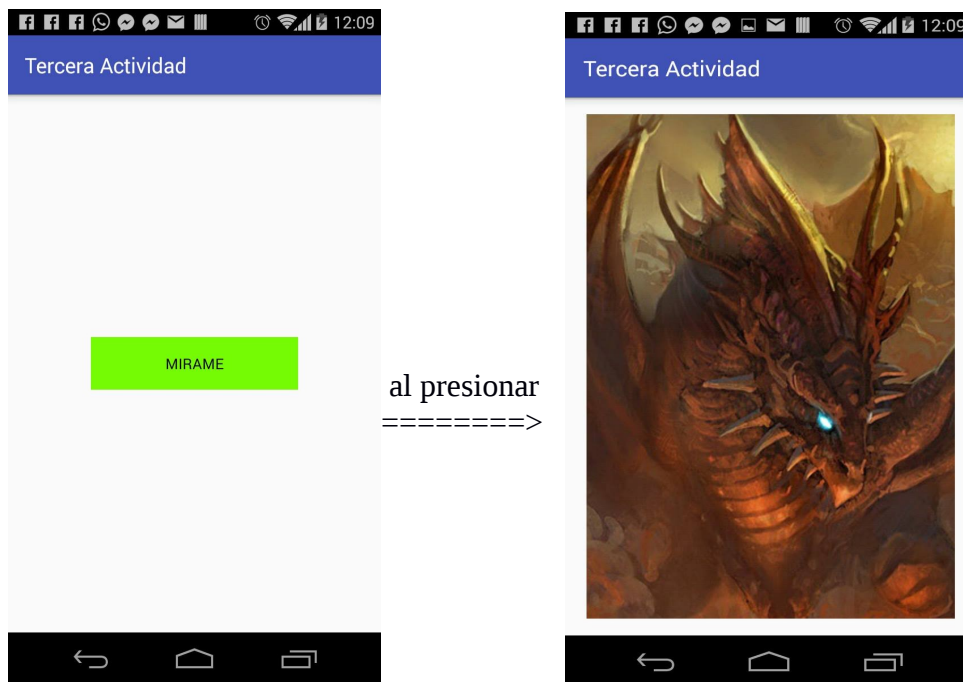
```
Intent intent= new Intent();
intent.setAction(Intent.ACTION_SEND);
intent.putExtra(Intent.EXTRA_TEXT, getResources().getString(
R.String.text));
intent.setType("text/plain");
startActivity(intent);
```



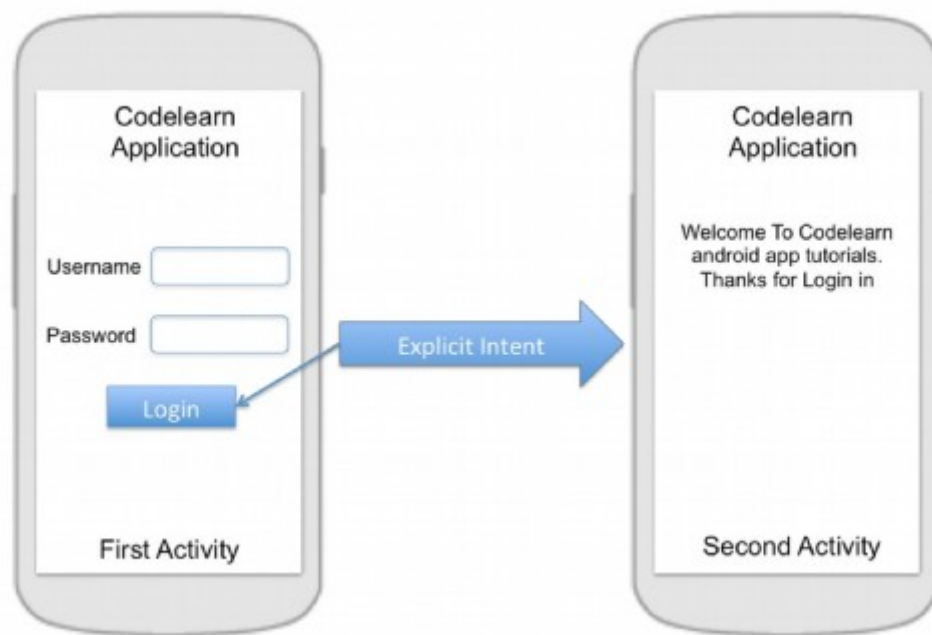
Practica

1. Una vez realizado el ejercicio de Intent, en el que un Button en la primera Activity nos lleva a la segunda y viceversa, se pide realizar:
 1. Dentro del anterior proyecto coloca en la segunda Activity un ImageView.
 2. Coloca un botón en la segunda Activity que sea capaz de cargar una imagen en el ImageView que acabamos de colocar.

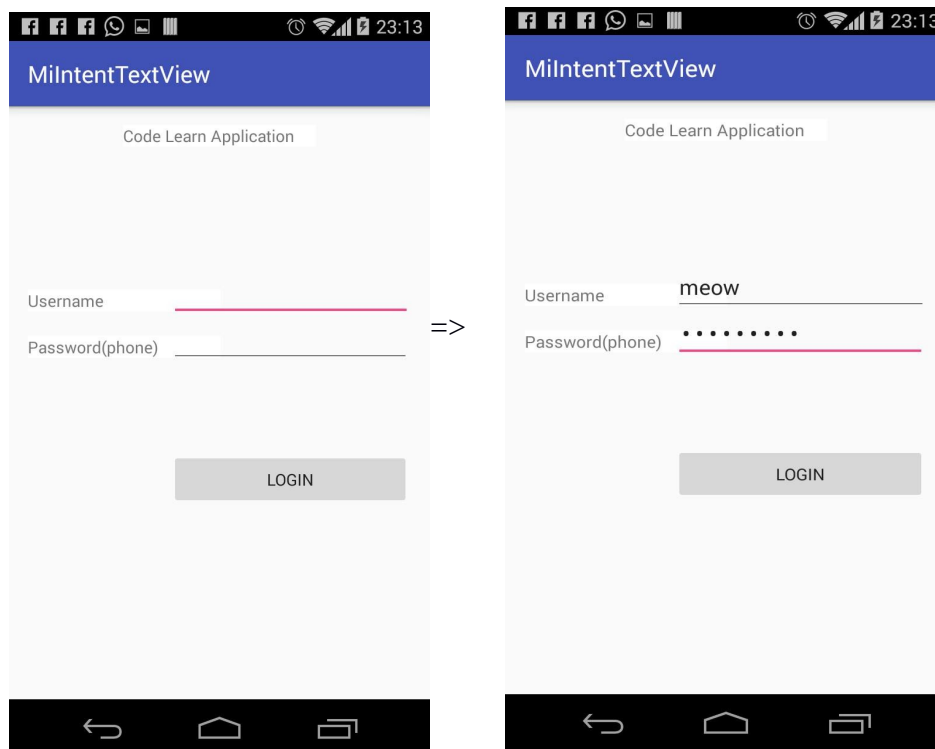
Modificando la tercera actividad ...

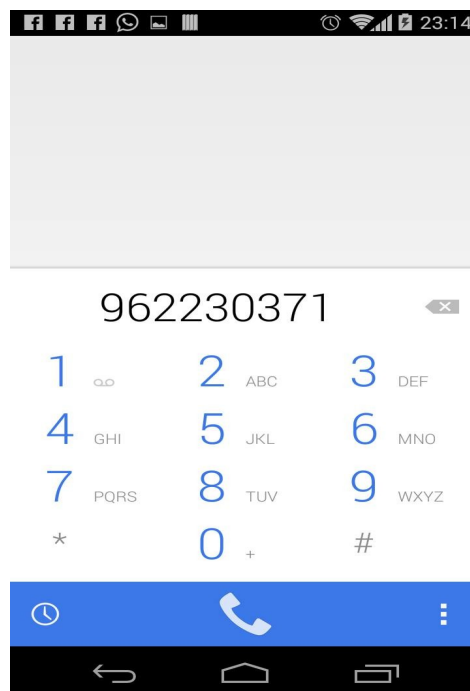
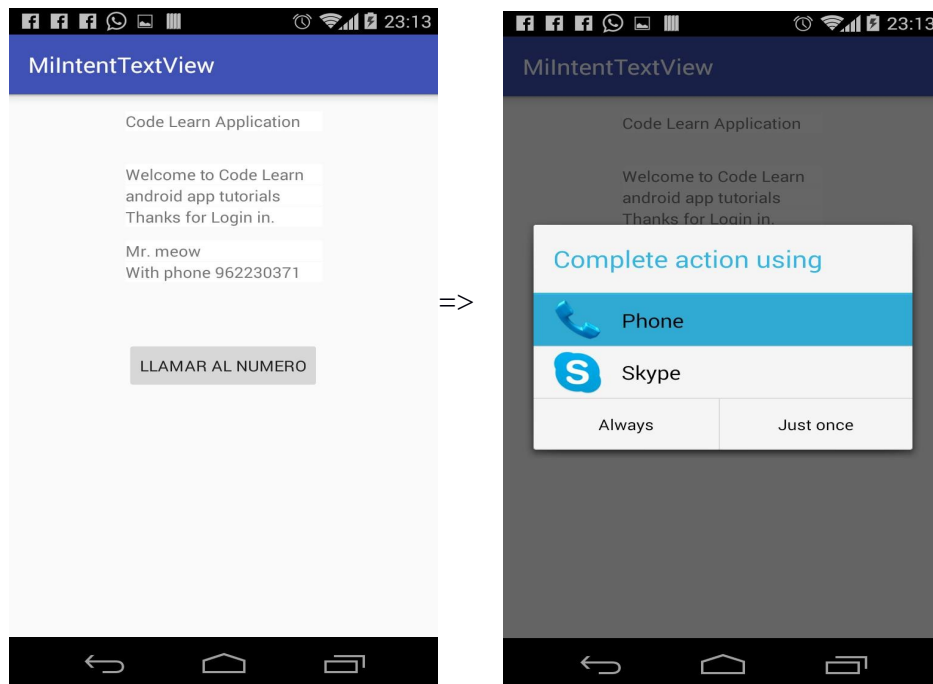


2. Crear una aplicación con dos actividades de manera que en la primera actividad nos pida un usuario y contraseña y nos lo muestre en la segunda actividad en un texto. Guiase del dibujo.



3. Realice una Intent que recoja de un TextView un número de teléfono y con una Intent derive a la actividad marcar el teléfono. Cualquier problema o comentario usar el foro de la asignatura.





Link del github con los códigos del laboratorio:

https://github.com/Jenazad/PDM/tree/master/Laboratorio_4

Referencias

<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=603>
<http://www.aprendeandroid.com/13/fundamentos3.htm>
<http://stackoverflow.com/questions/5089300/how-can-i-change-the-image-of-an-imageview>
<http://stackoverflow.com/questions/15842328/android-intent-action-call-uri>
http://www.tutorialspoint.com/android/android_phone_calls.htm
<https://developers.google.com/maps/documentation/android-api/intents>
<http://developer.android.com/intl/es/training/sharing/send.html>
<http://developer.android.com/intl/es/reference/android/app/AlarmManager.html>
<http://stackoverflow.com/questions/16256853/intent-to-launch-alarm-settings-in-clock-application-on-android>
<http://stackoverflow.com/questions/8517730/how-to-get-text-from-textview>
<http://stackoverflow.com/questions/1974489/how-to-get-text-from-edittext>