ESCUELA DE INGENIERÍA DE SISTEMAS
DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MOVILES
2016-II

TALLER # 2

Normas del Taller:

- ✓ El taller se desarrollará de manera individual.
- ✓ **Fecha de entrega Martes 4 de octubre por el Campus Virtual (no se recibe por ningún otro medio).**
- ✓ El objetivo del taller es abordar aspectos prácticos de conexión entre Activities de Android y el uso de aplicaciones nativas de Android (phone, Gmail, Line, Whatsapp, etc.)

Presentado Por: Juan Sebastian Rios Sabogal
Código: 1310105

1. Investigue el método createChooser() de la clase Intent.

   A. Adecue la app vista en la clase incorporando el uso de dicho método.

   B. Explique qué observa en el comportamiento de su app, al incorporar este método y describa las ventajas que tiene el uso de este método en comparación con la forma realizada en la clase.

R:*//* Para evidenciar el comportamiento del método createChooser() se ha modificado la app vista en clase incorporando dicho método, para esto he modificado el código de la clase CreateMessageActivity agregando los siguientes métodos:

Archivo: CreateMessageActivity.java

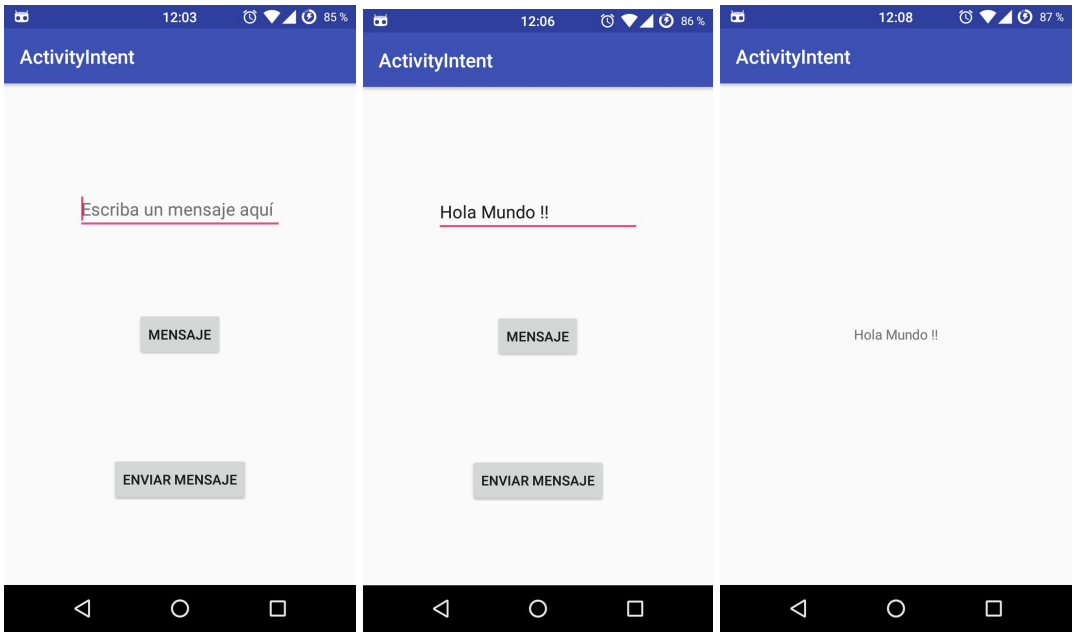
```
// Se llama este metodo cuando se hace click en el boton
public void onSendMessage(View view) {
   Intent intent = new Intent(this, ReceiveMessageActivity.class);
   EditText messageEdit = (EditText) findViewById(R.id.editText);
   String messageText = messageEdit.getText().toString();
   intent.putExtra(EXTRA_MESSAGE, messageText);
   startActivity(intent);
}


// Se llama este metodo cuando se hace click en el boton
```

```java
public void onCreateChooser(View view) {
    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.setType("text/plain");
    EditText messageEdit = (EditText) findViewById(R.id.editText);
    String messageText = messageEdit.getText().toString();
    intent.putExtra(Intent.EXTRA_TEXT, messageText);
    startActivity(Intent.createChooser(intent, "Titulo: ActivityIntent"));
}
```

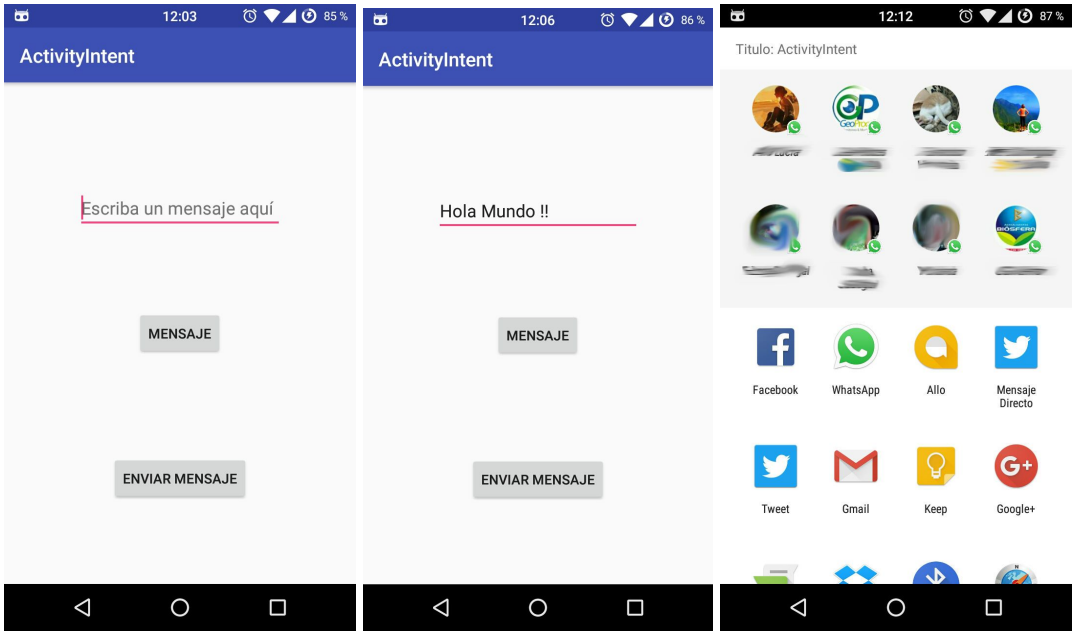
Cuando se activa el método `onSendMessage` al hacer click sobre el botón, este crea un objeto Intent y se configura para enviar el valor del mensaje escrito en un campo de texto hacia el Activity `ReceiveMessageActivity` que se encarga de mostrar el valor del mensaje recibido en un label.

Cuando se activa el método `onCreateChooser` al hacer click sobre el botón, este crea un objeto Intent pero con una acción de tipo enviar, se configura también el tipo de dato que se envía en el Intent y se hace uso del método createChooser() el cual delega la responsabilidad de seleccionar la Activity adecuada para recibir el mensaje al sistema operativo Android, dado el caso que existan más de una Actividad capaz de recibir el mensaje, se muestra un selector de aplicaciones al usuario para que él seleccione la aplicación que desea abrir para recibir el mensaje.

Evento del método `onSendMessage` en la aplicación:



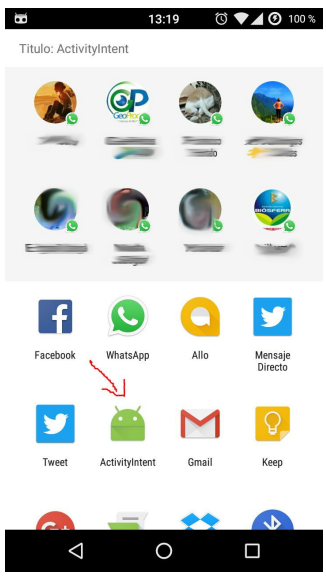
Evento del metodo `onCreateChooser` en la aplicación:



Si queremos que el Activity `ReceiveMessageActivity` este en capacidad de recibir mensajes y visible para otras aplicaciones del sistema Android, tenemos que agregar un filtro Intent en el archivo AndroidManifest.xml de nuestra aplicación.

Archivo AndroidManifest.xml

```
<activity android:name=".ReceiveMessageActivity">
   !-- This activity handles "SEND" actions with text data -->
   <intent-filter>
       <action android:name="android.intent.action.SEND"/>
       <category android:name="android.intent.category.DEFAULT"/>
       <data android:mimeType="text/plain"/>
   </intent-filter>
</activity>
```



De esta forma cuando hacemos uso del método createChooser el dialogo de selección creado por el sistema Android incorpora la Actividad `ReceiveMessageActivity` como una actividad que puede recibir el mensaje.

Solo tenemos que modificar la actividad `ReceiveMessageActivity` para filtrar los mensaje que llegan desde una actividad de la misma aplicación o como un mensaje recibido desde una actividad externa a la aplicación.

Archivo ReceiveMessageActivity.java

```
Intent intent = getIntent();
String messageText = intent.getStringExtra(EXTRA_MESSAGE);
if(messageText == null) {
   messageText = intent.getStringExtra(Intent.EXTRA_TEXT);
}
TextView messageView = (TextView) findViewById(R.id.message);
messageView.setText(messageText);
```

Algunas ventajas de usar el método createChooser() son:

- Si no existen aplicaciones que puedan recibir el mensaje, Android muestra un mensaje del sistema.
- Incluso si el usuario ha seleccionado una acción predeterminada para recibir el mensaje, se seguirá mostrando el selector.
- Se puede especificar un título para el diálogo de selección.

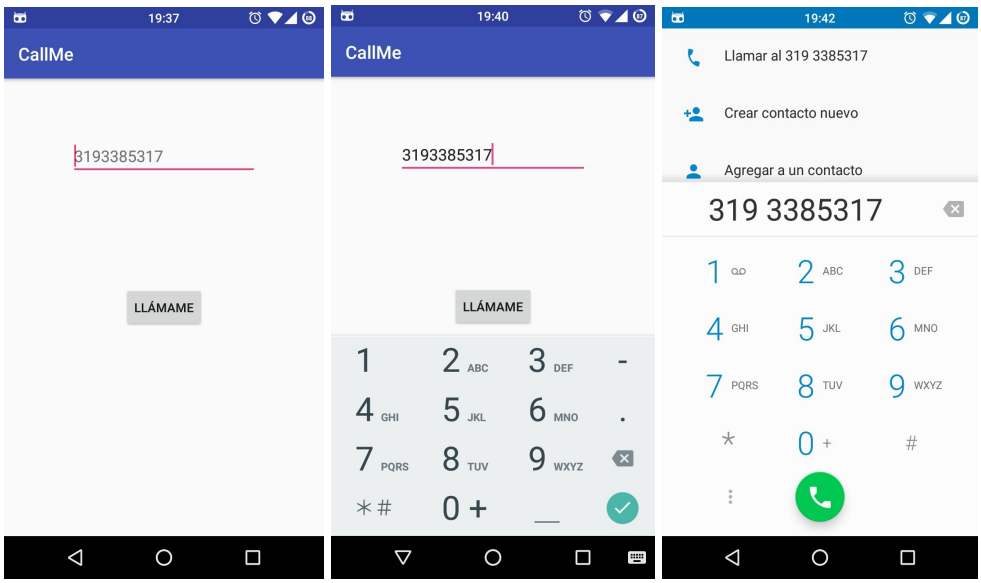
2. Usando como base la app desarrollada en clase, ahora adecuarla para que reciba un número telefónico y al pulsar el botón de enviar, se lance la aplicación de realizar llamadas telefónicas (Telefóno o Phone).

R://

Archivo: MainActivity.java

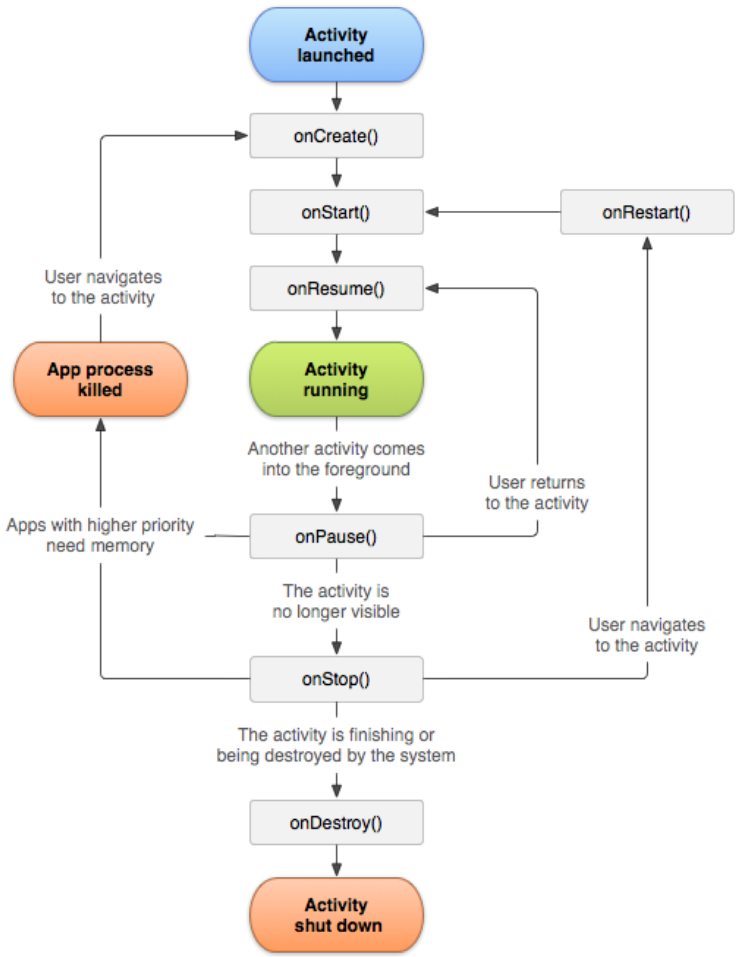
```
// Se llama este metodo cuando se hace click en el boton
public void onCallMe(View view) {
   EditText messageEdit = (EditText) findViewById(R.id.editText);
   String numberPhoneText = messageEdit.getText().toString();
   Uri number = Uri.parse("tel:" + numberPhoneText);
   Intent intent = new Intent(Intent.ACTION_DIAL, number);
   startActivity(intent);
}
```

Al hacer click sobre el botón, se obtiene el número de teléfono ingresado en el cuadro de texto, luego se crea un objeto Intent para ejecutar la aplicacion del telefono a la cual se envía el número de teléfono que el usuario ingresó.



3. Escriba en sus propias palabras el ciclo de vida de Activity en Android.

R://



El ciclo de vida de un Activity inicia cuando se crea la actividad con el método onCreate donde se inician las variables y se construye la vista, con el método onRestart se ejecuta después de que la actividad se ha detenido y antes de volver a iniciar, el método onStart se llama cuando la actividad es visible para el usuario, el método onResume es cuando la actividad está lista en la pila de actividades para recibir eventos del usuario, cuando el usuario abandona la actividad para ir a otra se ejecuta el método onPause, el método onStop se ejecuta cuando la actividad ya no es visible al usuario o porque otra actividad se ha reanudado, el método onDestroy se ejecuta cuando una actividad se ha finalizado o el sistema android decide terminar la actividad para reutilizar recursos.

Repositorio GitHub

El código fuente de la aplicación **ActivityIntent** y de la aplicación **CallMe** se puede descargar desde el siguiente link:

https://github.com/sebaxtian/Android-Apps