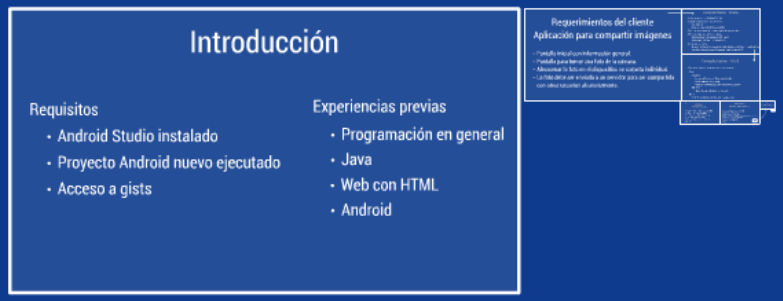
Android desde cero



diego.marcher.com.uy
diego@marcher.com.uy
Diego Marcher

Introducción

Requisitos

- Android Studio instalado
- Proyecto Android nuevo ejecutado
- Acceso a gists

Experiencias previas

- Programación en general
- Java
- Web con HTML
- Android

Requerimientos del cliente
Aplicación para compartir imágenes

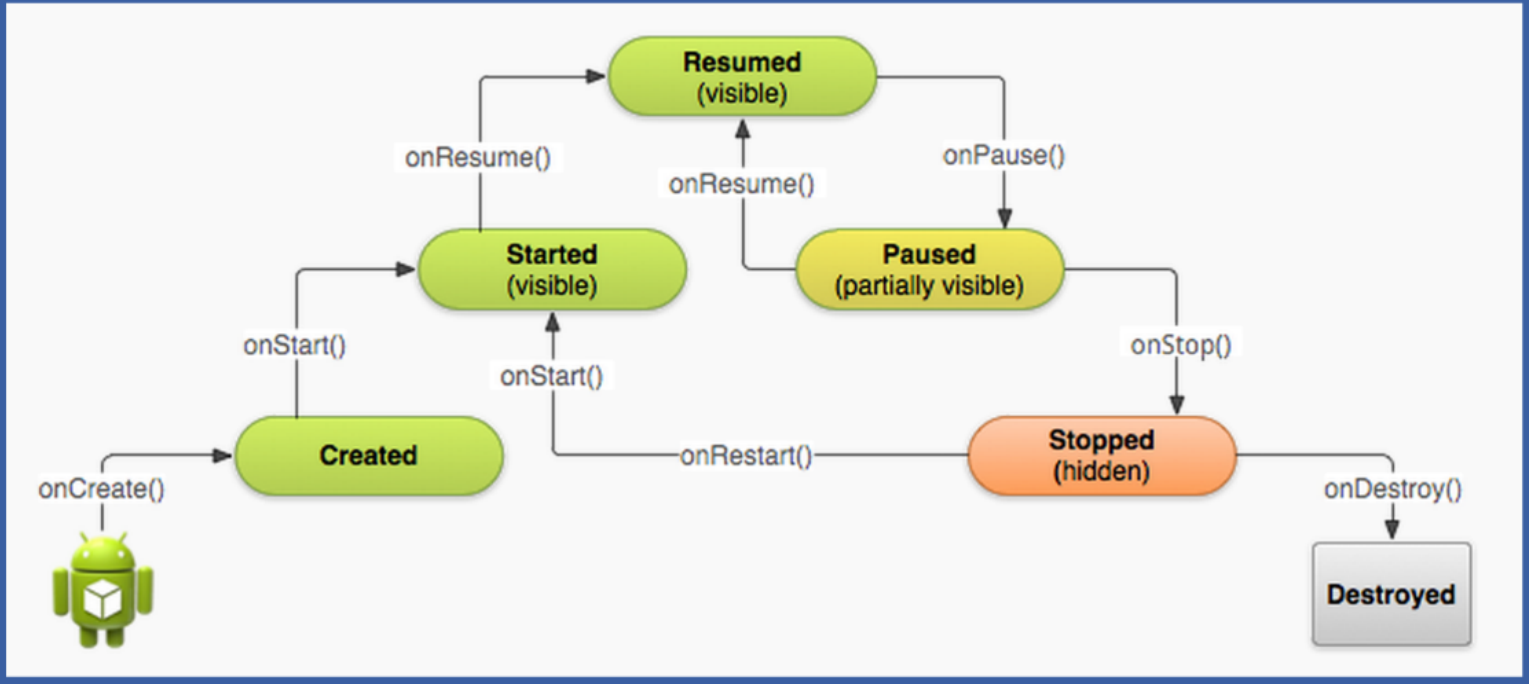
- Pantalla inicial con información general.
- Pantalla para tomar una foto de la cámara.
- Almacenar la foto en el dispositivo en carpeta individual.
- La foto debe ser enviada a un servidor para ser compartida con otros usuarios aleatoriamente.

Conceptos básicos - Activity

- Se declaran en el archivo Manifest.xml
- Pantalla compuesta por dos elementos
  - XML (layout)
  - Clase Java (controlador de actividad)
- Todo elemento/evento del layout controlado por una clase.
- Diferentes layouts según necesidades
  - RelativeLayout y LinearLayout (API level 1)
  - GridLayout (API level 14 (Android 4+))
- Decisiones de diseño
  - Elección de layouts y componentes (CalendarView API level 11 (Android 3+))
  - Uso de hardware (Cámara, Giroscópio, Acelerómetro, NFC)

Conceptos básicos - Intent

Ciclo de vida de una actividad



- Elección de layouts y componentes (CalendarView API level 11 (Android 3+))
- Uso de hardware (Cámara, Giroscópio, Acelerómetro, NFC)

Conceptos básicos - Intent

- Enviar mensajes de intención a otros componentes
- Tipos:
  - Implícito:
    - No especifica una actividad en particular.
    - Declara una acción general.
    - Otros componentes *escuchan* este evento y actúan.
  - Explícito:
    - Especifica la actividad a ser llamada.
- Extras:
  - Permiten enviar parámetros entre componentes

Intent explícitos

- Especifican la actividad a ser llamada.
- Requieren un contexto para invocar otras actividades
- Pueden enviar parámetros a través de extras.

```
Intent clientsIntent = new Intent(this, Clients.class);
clientsIntent.putExtra("clientId", client.getId());
startActivity(clientsIntent);
```


10:25
Clientes
Jayne
Ball Corporation
Lyle
PNM Resources Inc
Rodrick
NVR Inc
Darrin
Metaldyne Corporation
Jade
Anadarko Petroleum Corporation
Quiana
Regis Corporation
Tiffani
Lockheed Martin Corporation
Elba
SPX Corp


10:30
Rodrick
NVR Inc
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed id viverra libero, non tristique mi. Duis congue viverra mauris. Integer lorem tellus, tincidunt ac mauris eget, imperdiet venenatis nulla. Suspendisse in mollis eros. Morbi vel fringilla urna. Duis cursus vel lectus non convallis. Curabitur fermentum eu sem semper auctor. Aliquam rhoncus maximus sem, nec finibus sem tempus eu. Quisque nec varius tortor. Cras imperdiet cursus augue id commodo. Mauris blandit nisi tempus, rhoncus mi quis, accumsan felis.
Praesent vulputate, metus nec finibus laoreet, lacus nibh vulputate mi, ac rhoncus mauris arcu eget tellus. In viverra tellus id risus condimentum pellentesque. Proin quis libero mattis, suscipit ligula eu, lobortis lectus. Mauris eu turpis tortor. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Morbi sagittis scelerisque erat, id eleifend erat posuere at. Curabitur interdum sagittis felis, eu semper quam blandit sed. Sed sed consectetur sem. Praesent ante

• Permiten enviar parámetros en

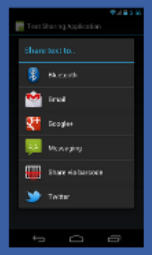
Taller parte 1
Pantallas (layouts)

- Creación de nueva actividad ***CameraActivity***.
- Crear TextView en ***CameraActivity***.
- En actividad principal, botón ***Tomar foto***.
- Crear handler de evento ***onClick***.
- Linkear actividades a través de un Intent.
- Ejecutar.

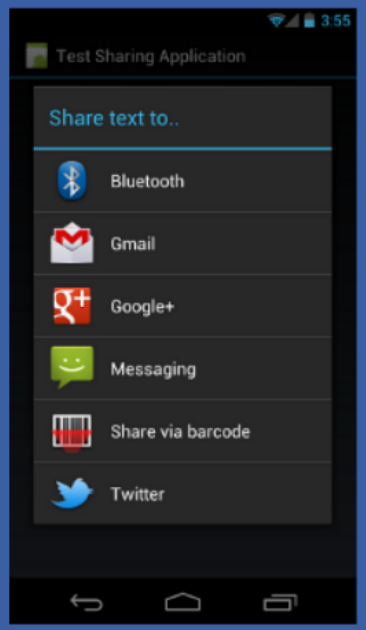
Intent implícito

- No especifica una actividad en particular.
- Declara acción general.
- Otros componentes se registran para escuchar este evento.
- Pueden enviar parámetros a través de extras.
- Generalmente se espera un callback en método *onActivityResult*.

```
Intent intent = new Intent(Intent.ACTION_SEND);
String title = "Share text to..";
Intent chooser = Intent.createChooser(intent, title);
startActivity(chooser);
```



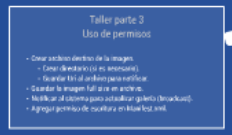
```
Intent intent = new Intent(Inter
String title = "Share text to..
Intent chooser = Intent.createCh
startActivity(chooser);
```



ntre componentes

Taller parte 2
Uso de hardware (cámara)

- En ***CameraActivity***, botón de acción ***Tomar Foto***.
- Crear handler del evento ***onClick***.
- Crear ImageView como placeholder.
- Invocar cámara con Intent implícito ***MediaStore.ACTION_IMAGE_CAPTURE***.
- Crear método de callback, ***onActivityResult***.
- Cargar imagen en el placeholder.
- Agregar permiso en ***Manifest.xml***.
- Ejecutar.



Taller parte 3
Uso de permisos

- Crear archivo destino de la imagen.
  - Crear directorio (si es necesario).
  - Guardar Uri al archivo para notificar.
- Guardar la imagen full size en archivo.
- Notificar al sistema para actualizar galería (broadcast).
- Agregar permiso de escritura en Manifest.xml.

Consumo de servicios

Requiere permiso para acceder a Internet

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

Opcionalmente para obtener el estado de la red:

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

Es necesario ejecutar la llamada desde un hilo que no sea el de la UI.

Hilos que no son el de UI no pueden modificarla.

Método helper en clase Activity

- runOnUiThread(Runnable action)

Consumo de servicio con Thread

```
Thread thread = new Thread() {
    @Override
    public void run() {
        /** Invocación a servicio **/
        serverCall();

        runOnUiThread(new Runnable() {
          @Override
          public void run() {
            /** Actualizar la UI **/

            /** Posibles problemas :) **/

          }
        });
    }
};
thread.start();
```

Invocación de servicio con AsyncTask

```
new AsyncTask<Params, Progress, Result>() {

    @Override
    protected Result doInBackground(Params... params) {
       /** Invocación a servicio **/
       return null;
    }

    @Override
    protected void onProgressUpdate(Progress... values) {
       /** Notificar UI del progreso **/
    }

    @Override
    protected void onPostExecute(Result result) {
       /** Actualizar la UI **/
    }

}.execute();
```

Taller parte 4
Mock de invocación de servicio con AsyncTask

- En **CameraActivity** crear botón Enviar Foto.
- Agregar TextView para mostrar progreso.
- Crear handler de onClick llamado enviarFoto.
- Crear AsyncTask con 3 métodos
  - doInBackground (mock de servicio).
  - onProgressUpdate (actualiza progreso en la UI).
  - onPostExecute (notifica UI de fin).
- Agregar permisos al Manifest.
- Ejecutar.

Gracias