

Práctica de computación ubicua

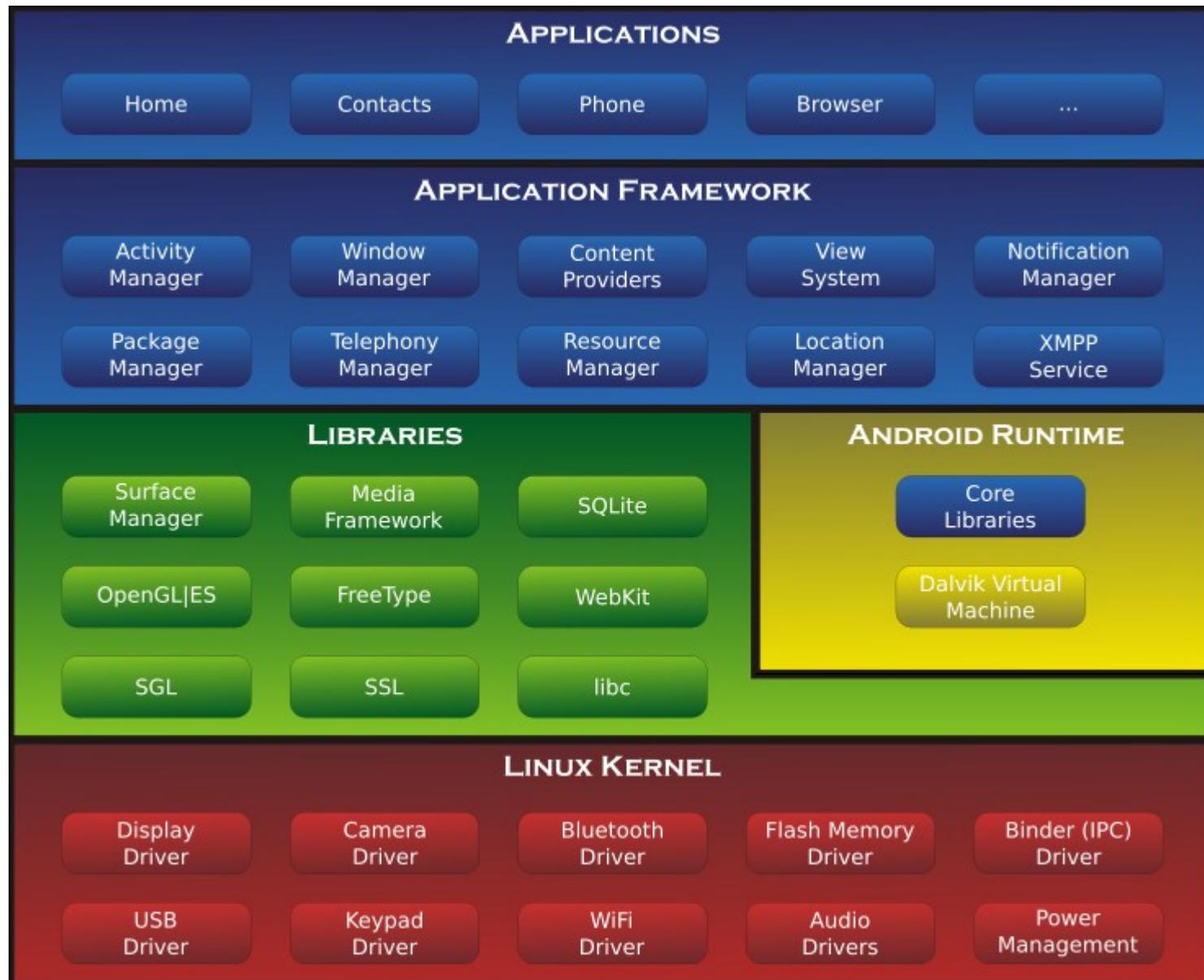
Aplicación android context-aware

Fernando Pérez Costoya
fperez@fi.upm.es

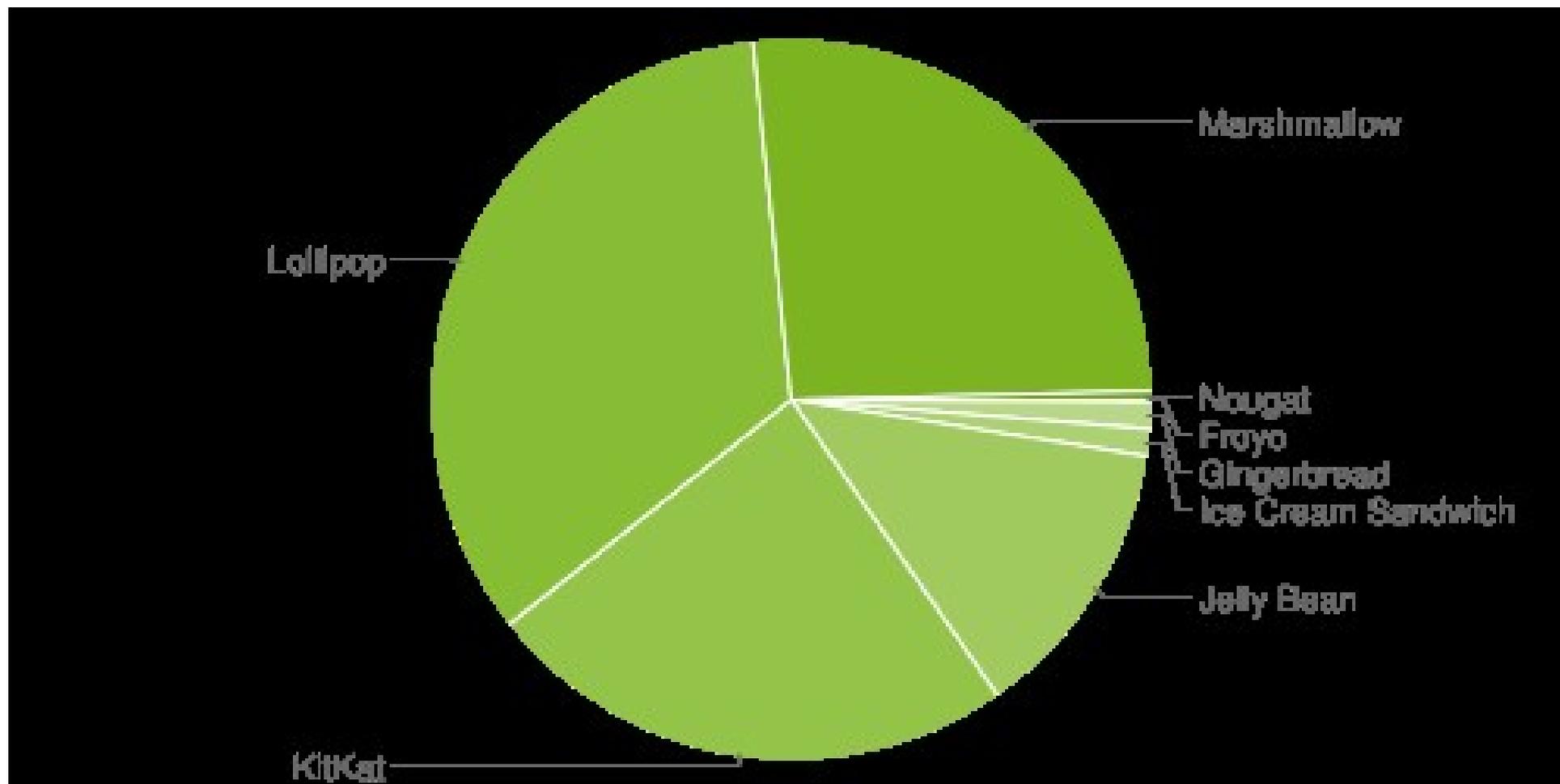
Introducción

- Sistema operativo para móviles
 - Basado en Linux
 - Cada *app* tiene su propio UID
- Entorno de desarrollo de *apps* móviles
 - Basado en Java
 - IDE recomendado: Android Studio
- Un poco de historia:
 - Android Inc. (2003); comprado por Google (2005); recomendado por *Open Handset Alliance* (2007)
 - Versión actual: 7.0 *Nougat* (API 25)
- Dominante en mercado de móviles

Pila software de Android (wikipedia)



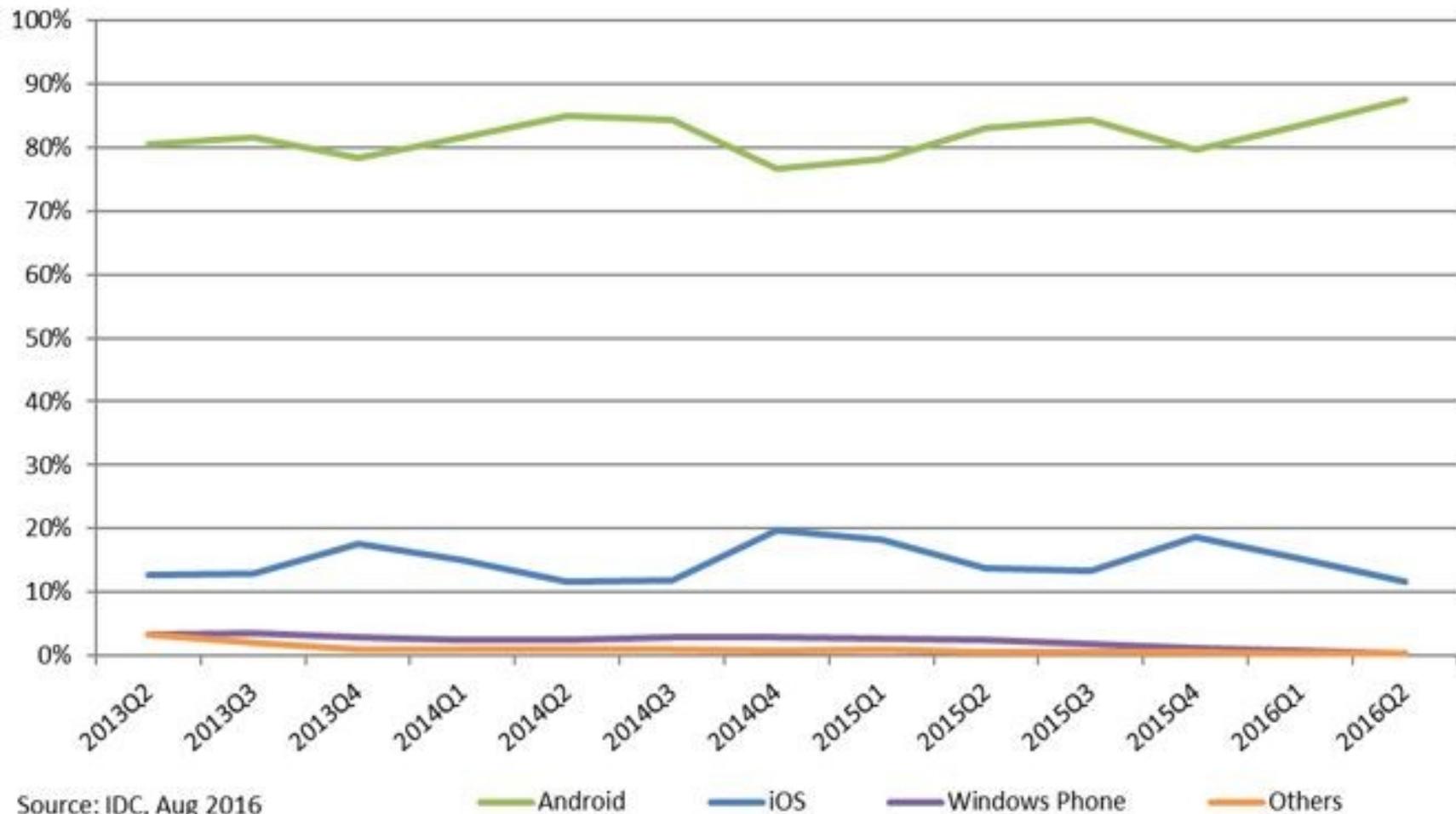
Cuota mercado versiones de Android



<https://developer.android.com/about/dashboards/index.html>

Cuota mercado SSOO para móviles

Worldwide Smartphone OS Market Share
(Share in Unit Shipments)



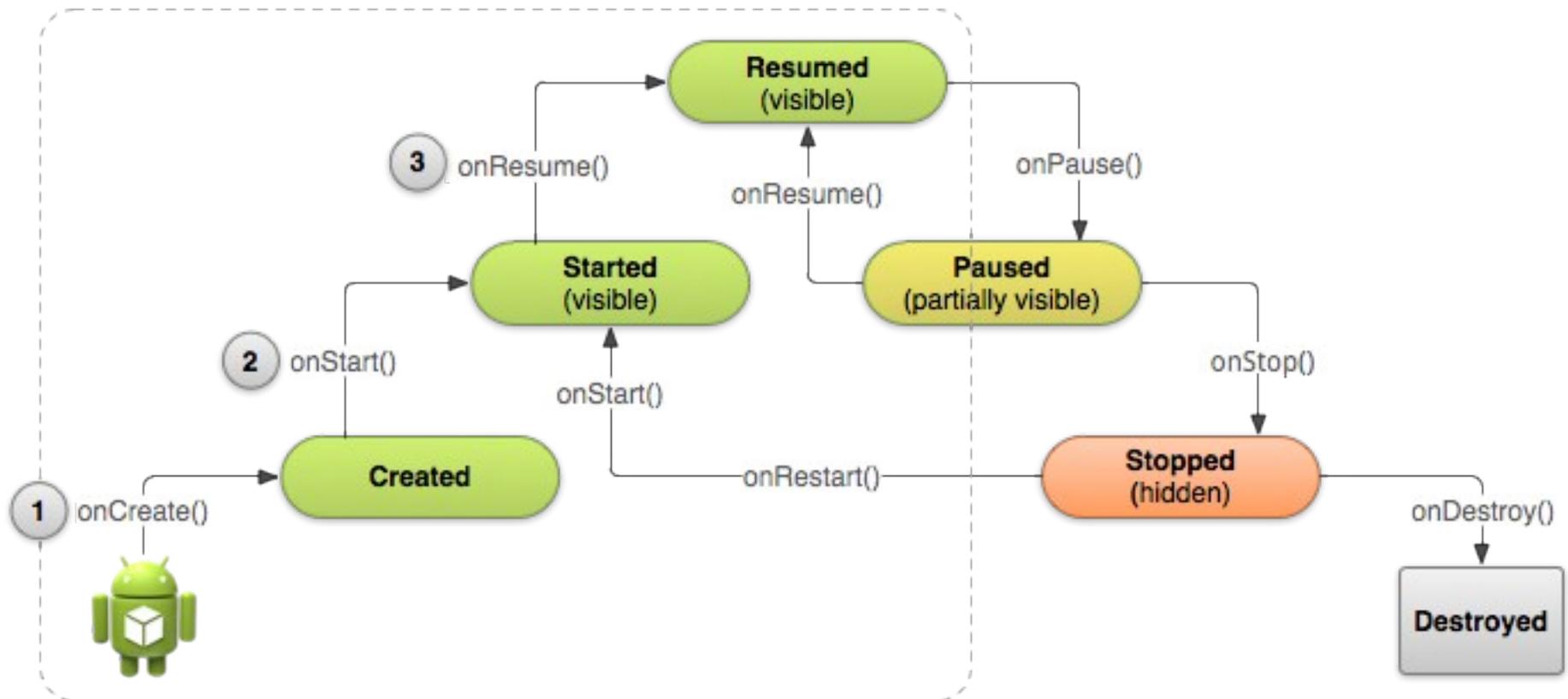
Source: IDC, Aug 2016

<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

Componentes

- *App* puede tener 4 tipos de componentes:
 - Actividades (*Activities*):
 - Controlador de la UI; 1 actividad por pantalla
 - Servicios (*Services*):
 - Trabajo en segundo plano sin UI
 - P.e. Reproducción de audio en segundo plano
 - Proveedores de contenido (*Content Providers*)
 - Proveen a *apps* acceso a datos compartidos
 - P.e. Agenda de contactos
 - Receptores de multidifusión (*Broadcast Receivers*)
 - Responden a *broadcasts* del sistema o de otras *apps*
 - P.e. Señal de batería baja

Ciclo de vida de una actividad



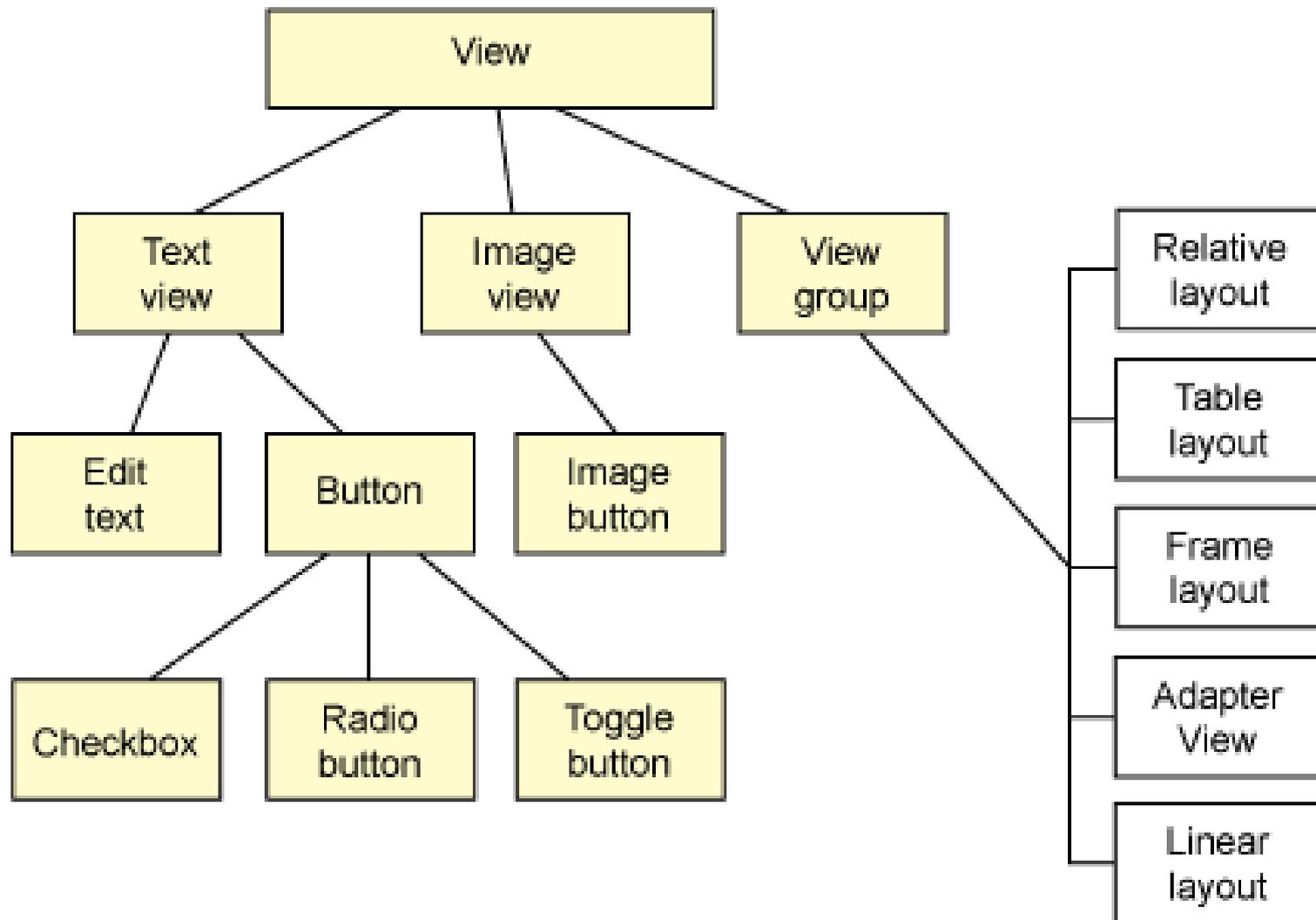
Componentes

- Declarados en el fichero *Manifest*
 - Junto con otra metainformación de la *app*
 - Permisos requeridos, HW requerido, bibliotecas usadas, API mínimo y *target* (actualmente, en fichero de proyecto),...
- Se comunican mediante *Intents* (“mensajes”)
- Por defecto, ejecución en mismo *thread*
 - Todos los componentes ejecutados en mismo *thread*
 - Operación larga o bloqueante: UI no responde
 - Usar un modo asíncrono:
 - Se inicia operación y se recibe aviso de cuando termina
 - Crear *threads* adicionales

Interfaz de usuario

- Similar a cualquier GUI
 - Objetos contenedores: ViewGroup
 - FrameLayout, RelativeLayout, LinearLayout, TableLayout, GridLayout, ScrollView, ListView,...
 - Objetos de diálogo: View
 - TextView, Button, ImageButton, EditText,...
 - Programador puede crear nuevos
- App puede incluirlos “programáticamente”
 - Pero mejor en fichero XML externo
 - Separación vista y controlador

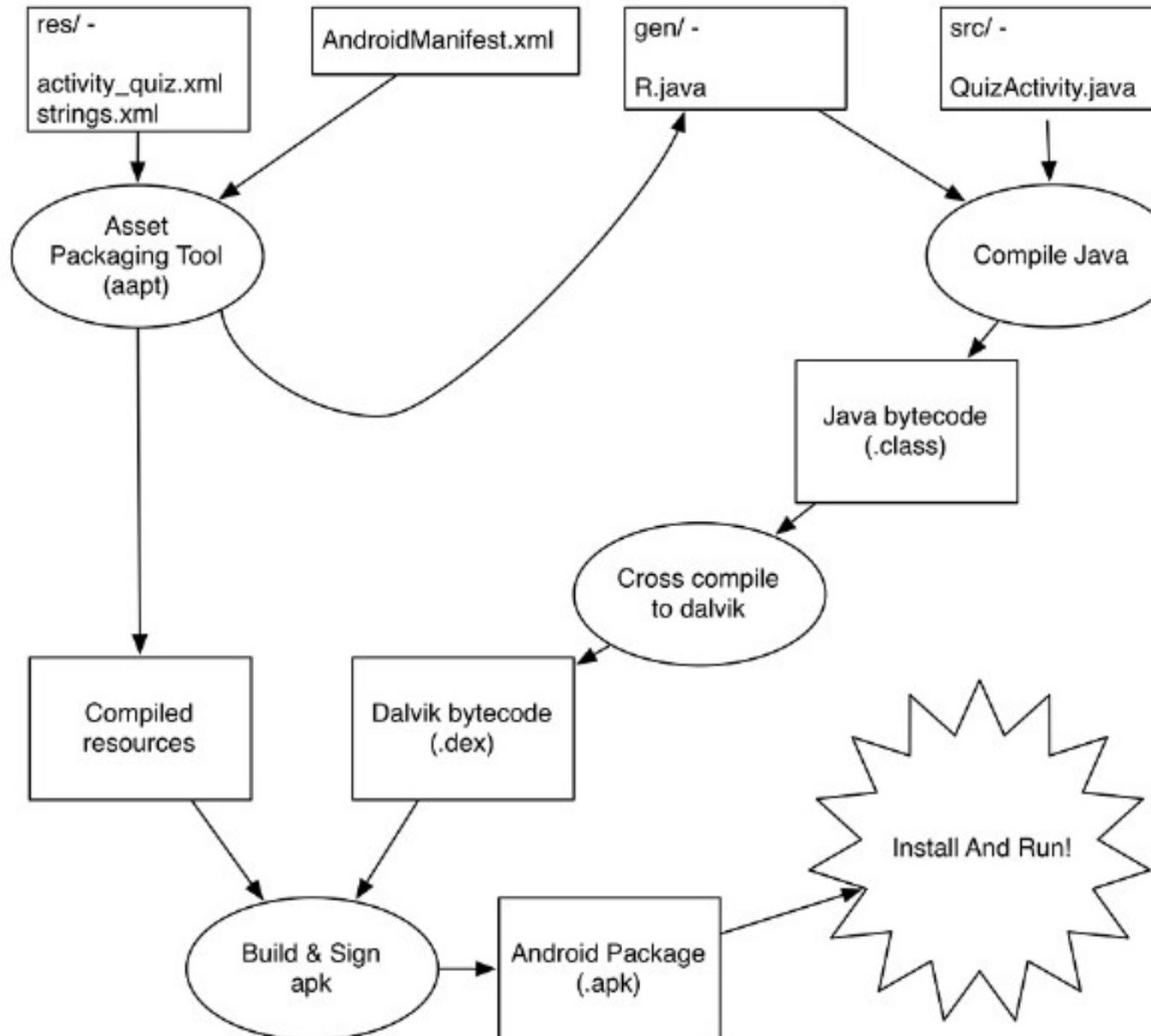
Árbol de Views (incompleto)



Recursos

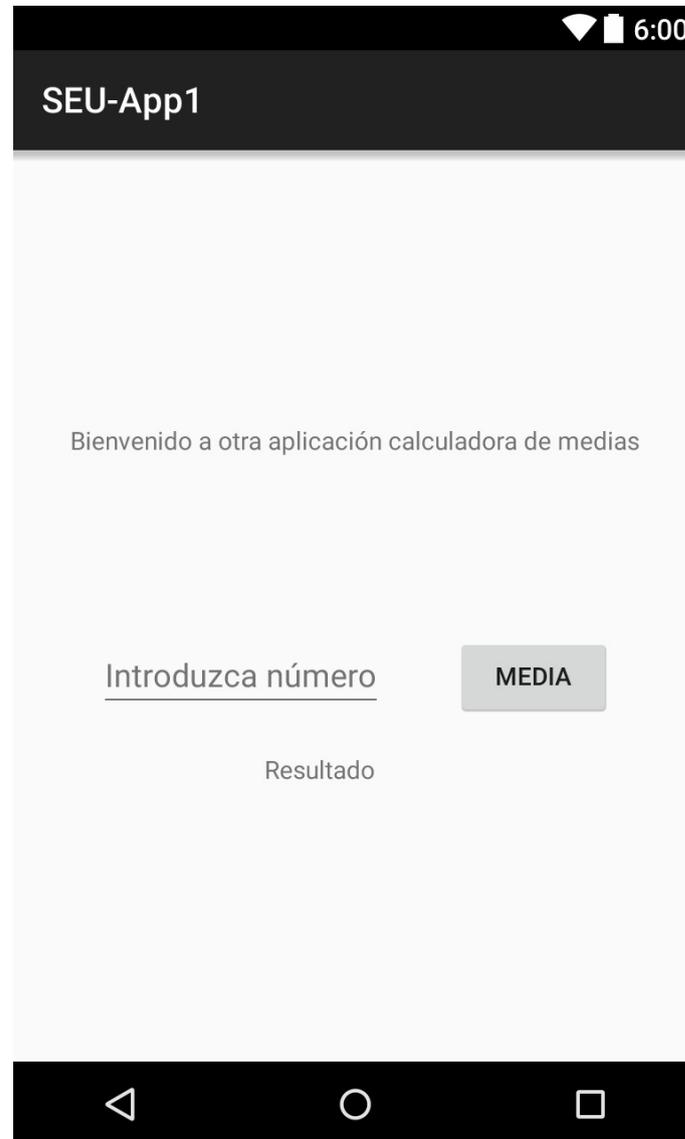
- Aplicación debe “externalizar” recursos
- Subdirectorios de directorio *res*:
 - *layout* (diseño GUI), *values* (*strings*, colores, dimensiones, estilos,...), *drawables*,...
 - Se empaquetan junto al código de la *app*
- Se pueden definir valores defecto y alternativos
 - Permite adaptación automática de *apps*
 - Directorios con sufijos calificadores de configuración:
 - P.e. *Values-es*, *layout-land*, *values-es-land*,...
 - Orden de calificadores expresa su prioridad:
 - <http://developer.android.com/guide/topics/resources/providing-resources.html>
 - Algoritmo busca mejor encaje

Proceso de generación de App



Primera aplicación

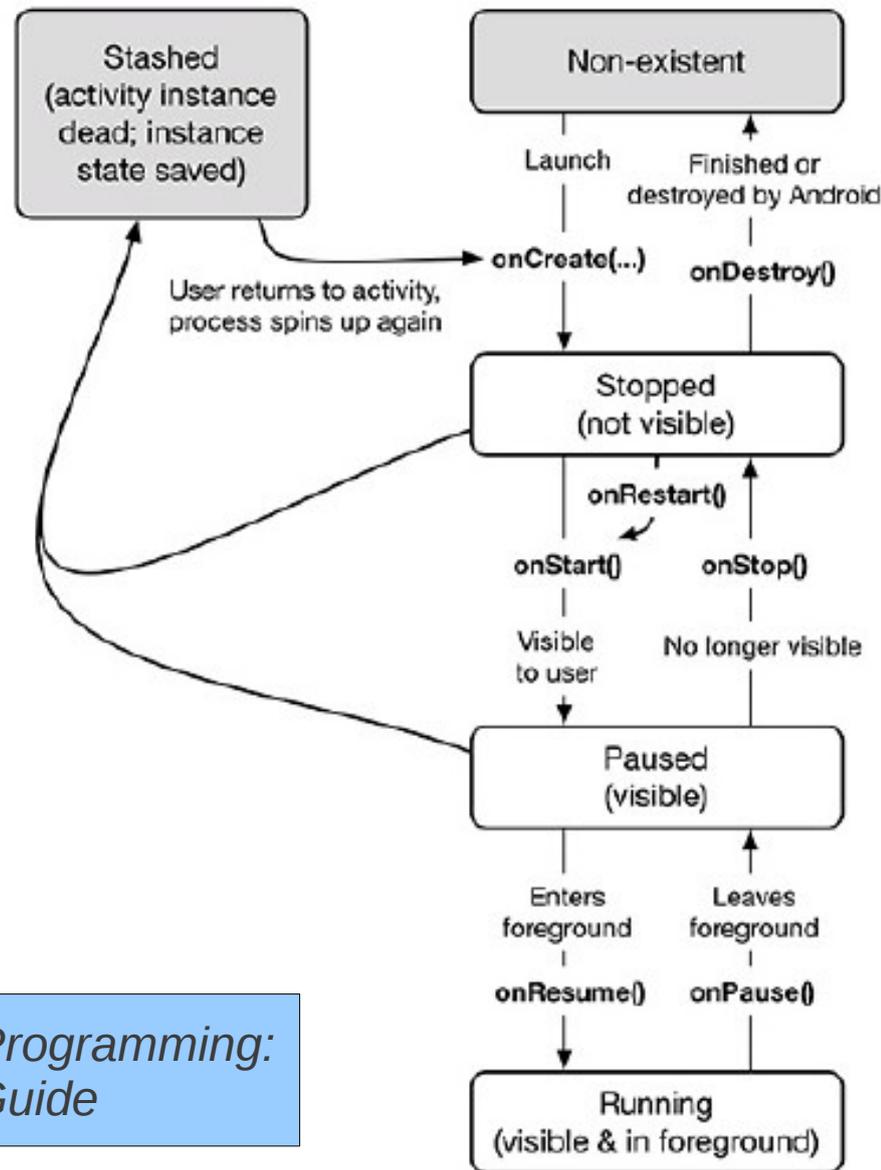
Prueba a rotar el dispositivo (Ctrl-F12).
¿Qué sucede?
¿Cómo se arregla?



La frágil memoria de las actividades

- En el ejemplo, si se rota dispositivo:
 - Se mantiene info. de UI pero se pierde la de la app
 - Lo mismo cambiando *locale*
 - Juega ahora con botones *Back*, *Home*, *Recents*
- Si Android destruye actividad
 - Por cambio de configuración
 - Por falta de recursos destruye proceso que la contiene
 - Orden: 1º proceso vacío; 2º *background*; 3º visible; 4º *foreground*;
 - No por *finish* o botón *Back*
 - Salva estado de UI y lo restaura al reactivarse
 - El resto de estado se pierde

Ciclo de vida *revisitado*

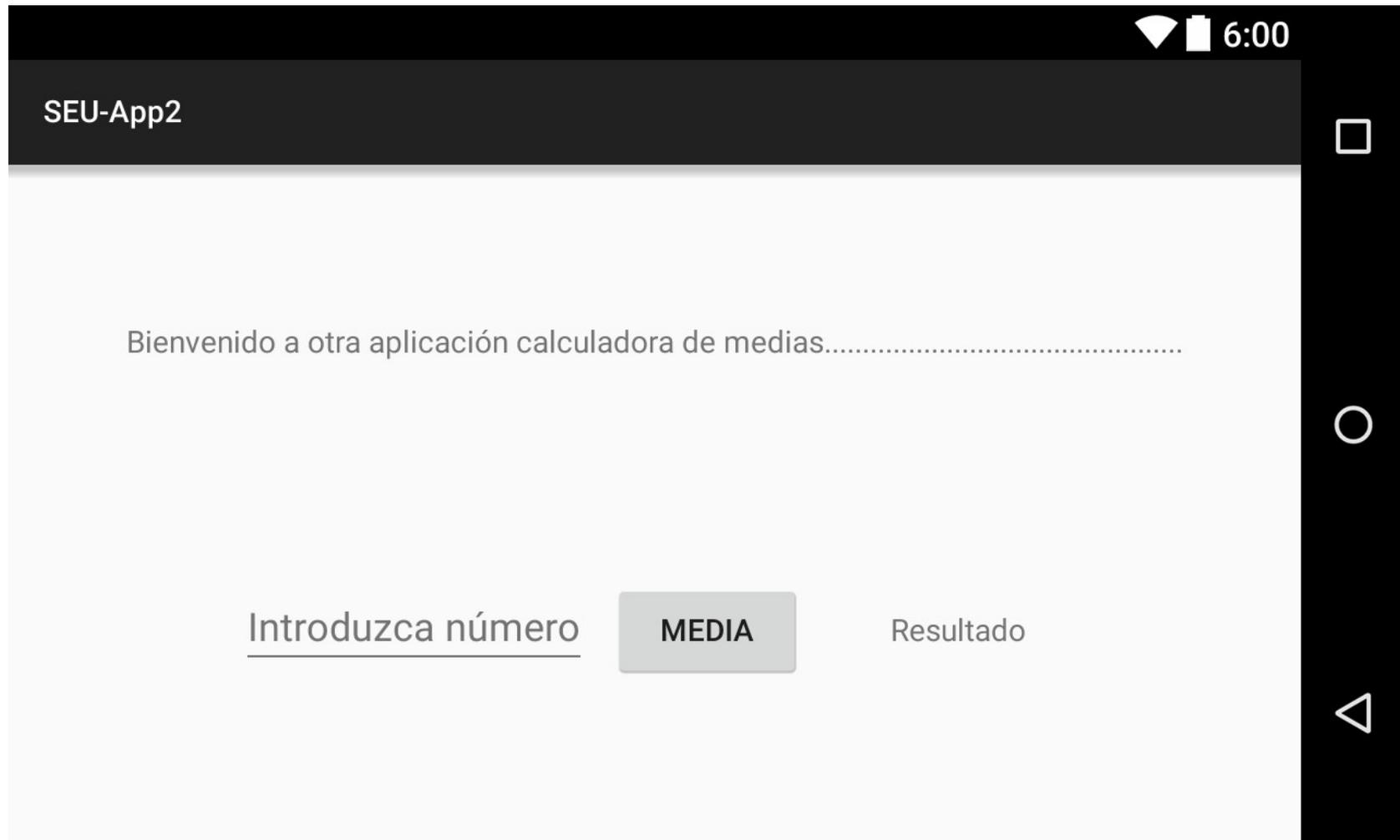


Extraído de: *Android Programming: The Big Nerd Ranch Guide*

Salvando el estado de una actividad

- Activ. destruida por cambio conf o falta recursos
 - SO invoca: *onSaveInstanceState(Bundle)*
 - Se almacena en *Bundle* par: ID variable a salvar y su valor
 - *putString, putBoolean,...*
- Actividad recreada: *onCreate(Bundle)*
 - Se recuperan las variables (*getString, getBoolean,...*)
- Alternativa: gestión de cambios de configuración
 - No destruye actividad; invoca *onConfigurationChanged*
 - En manifiesto debe incluir *android:configChanges*
 - *android:configChanges="orientation"*

2ª app: 1ª app con memoria y adaptada a orientación y lenguaje



Apps asociadas a la práctica

- App3:
 - *Broadcast receiver*: cambios nivel batería
- App4:
 - Gestión de sensores
- App5:
 - Localización