Desarrollo de Aplicaciones Móviles Android

PRESENTACIONES

- Pablo Formoso Estrada
- Licenciado en Informática en la UEM
- Master en Dirección estratégica y gestión de la innovación por la IUP
- Muy social
  - @pabloformoso
  - LinkedIn, Klout, CoderWall, Github, About.me, XIng, Quora
  - Blog personal pabloformoso.com
  - Cañas y tapas =)
- Fundador de Softwhisper, empresa dedicada a la movilidad.

citic

SoftWhisper

EL CURSO

1. Introducción al entorno de Android
2. Estructura y planificación del desarrollo de una app
3. Fragments e Intents (también broadcast receivers)
4. Operaciones de red
5. Gestión y acceso al sistema de ficheros
6. Acceso al hardeware del dispositvo
7. Componentes multimedia
8. Geocoding y mapas
9. SDK "Avanzada"
10. EXTRA SHOT: Processing. Rapid Android Development.

citic

SoftWhisper

INTRODUCCIÓN AL ENTORNO

citic

SoftWhisper

¿QUE ES ANDROID?

- Plataforma abierta para el desarrollo de aplicaciones ¿móviles?
  - Sistema operativo de alto nivel, Android.
  - SDK a dos niveles, desarrollo de apps y NDK
  - Teléfonos, tablets, TV y sistemas empotrados.
- Inspirado en los replicantes de BladeRunner (Nexus-6) de ahi su nombre, por el famoso Andy de la novela.

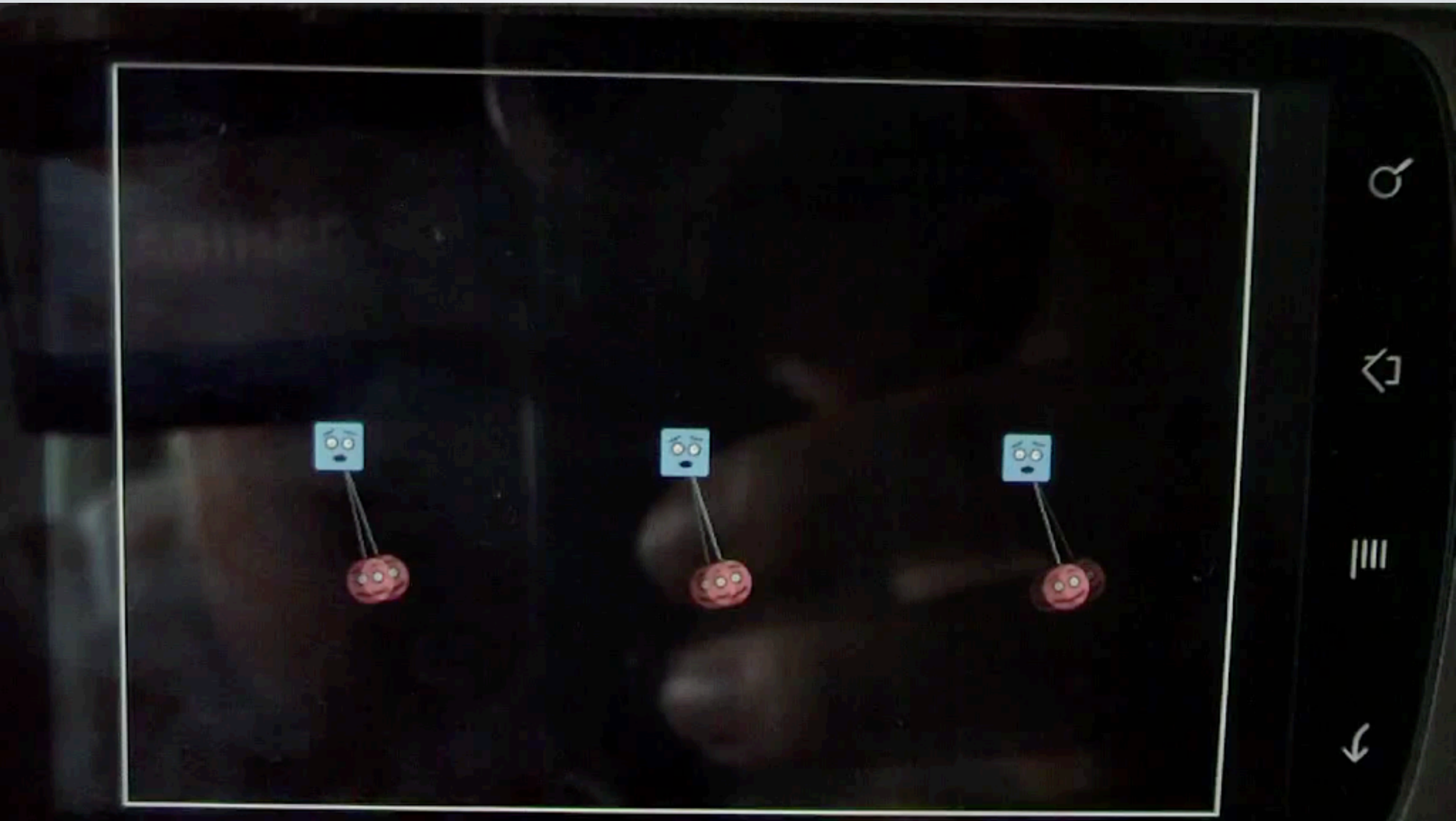
...Y MÁS EN DETALLE

- Un kernel de Linux para interfaces a bajo nivel
- Mucho OpenSource
  - SQLite, Webkit, OpenGL ES (1.0 y 20) ...
- Maquina virtual Dalvik (o ART) para la ejecución de Apps
- Frameworks de desarrollo a alto nivel
- Framework especialidado para la UI de usuarios
- Conjunto de herramientas para el desarrollo
  - ADT, ADB, DDMS

FUNCIONALIDADES TÍPICAS

- Consumo de servicios/APIs para información
- Integración de elementos multimedia
- Geolocalización y posicionamiento en mapas
- Servicios en segundo plano
- Widgets
- Juegos 2D/3D a traves de GL ES o AndEngine.
- Chorriapps...

AND ENGINE FÍSICAS



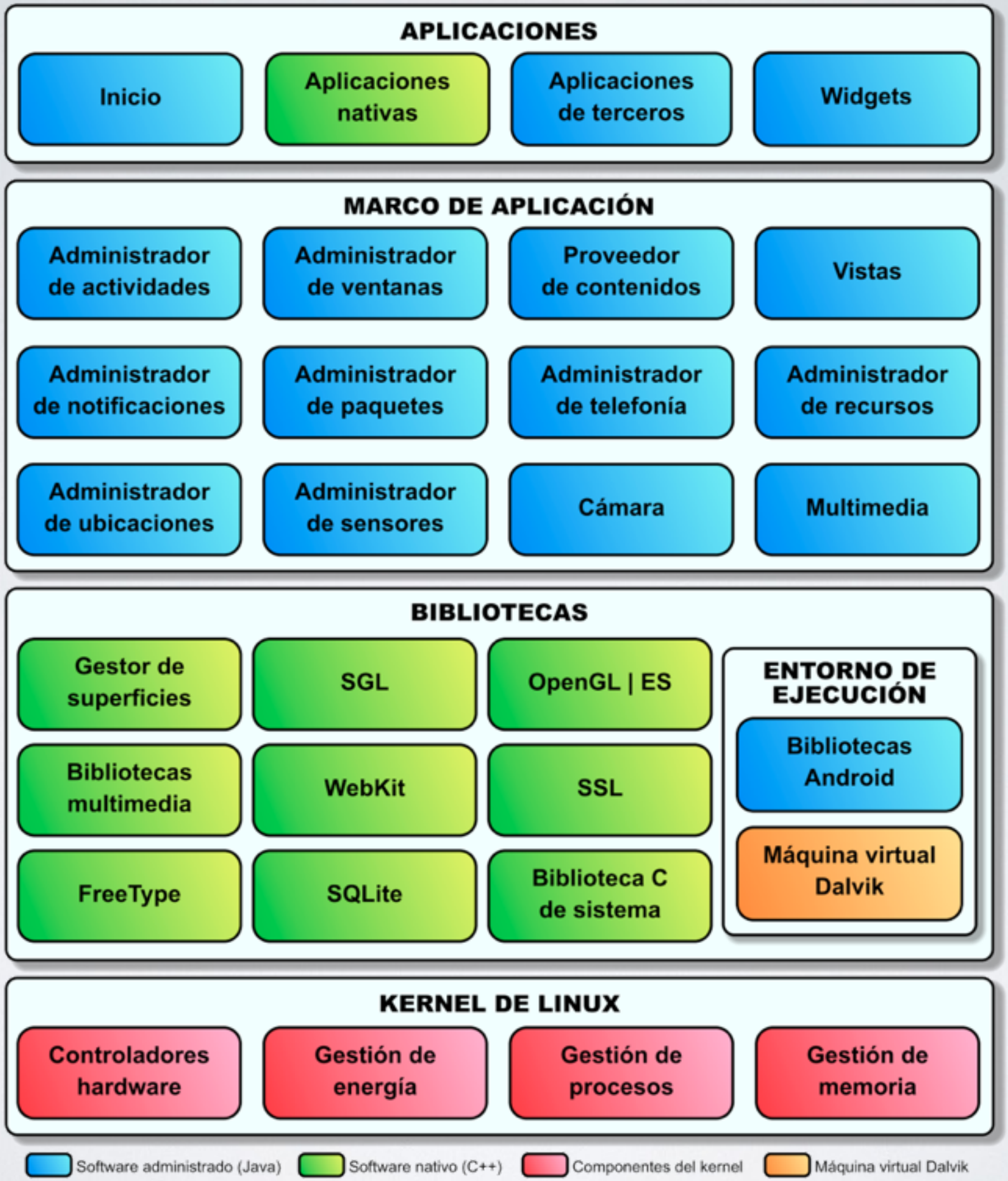
USOS MÁS DIVERTIDOS

- Conexiones NFC
- AR y vision por comutador (QRs, Datamatrix, Barcodes)
- Conexiones Wifi-Direct
- Integración de periféricos
  - Vía bluetooth
  - Usando cables usb-mini
- Compilación de shaders
- Aplicaciones médicas
- Binomio Ekahau y Android para posición en interiores

1.1 SDK DE ANDROID

citic

SoftWhisper


APLICACIONES
Inicio
Aplicaciones nativas
Aplicaciones de terceros
Widgets
MARCO DE APLICACIÓN
Administrador de actividades
Administrador de ventanas
Proveedor de contenidos
Vistas
Administrador de notificaciones
Administrador de paquetes
Administrador de telefonía
Administrador de recursos
Administrador de ubicaciones
Administrador de sensores
Cámara
Multimedia
BIBLIOTECAS
Gestor de superficies
SGL
OpenGL | ES
Bibliotecas multimedia
WebKit
SSL
FreeType
SQLite
Biblioteca C de sistema
ENTORNO DE EJECUCIÓN
Bibliotecas Android
Máquina virtual Dalvik
KERNEL DE LINUX
Controladores hardware
Gestión de energía
Gestión de procesos
Gestión de memoria
Software administrado (Java)
Software nativo (C++)
Componentes del kernel
Máquina virtual Dalvik

INSTALACIÓN

- Linux, OS X 10.6 + o Windows.
- Java SDK (no vale JRE)
  - 1.6 +
    - http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html
- Android SDK
  - http://developer.android.com
- Eclipse Indigo o superior, versión clásica.

INSTALACIÓN ADT

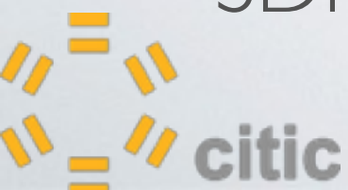
- Android Development Tools for Eclipse
  - http://developer.android.com/tools/sdk/eclipse-adt.html
- Gestor de emuladores integrado en Eclipse
- Actualizaciones de la SDK directamente desde Eclipse
- Provee de códigos de ejemplo y APIs de terceros en algunas versiones.

COMPONENTES SDK

- ADB (Android Debuger Bridge)
  - Encargado de gestionar la conexión de los dispositivos
  - adb start-server
  - adb kill-server
  - adb get-state
- AVD Manager
  - Creación y gestión de los emuladores de dispositivos.

MÁS SDK

- Documentación y ejemplos
- Código fuente de la SDK
- APIs de Google
- Soporte de compatibilidad hacía atrás. Compatibility mode.
- Gestión de cobros en Google Play
- Comprobación de licencias
- Generación y firma de las apps
- SDK Tools

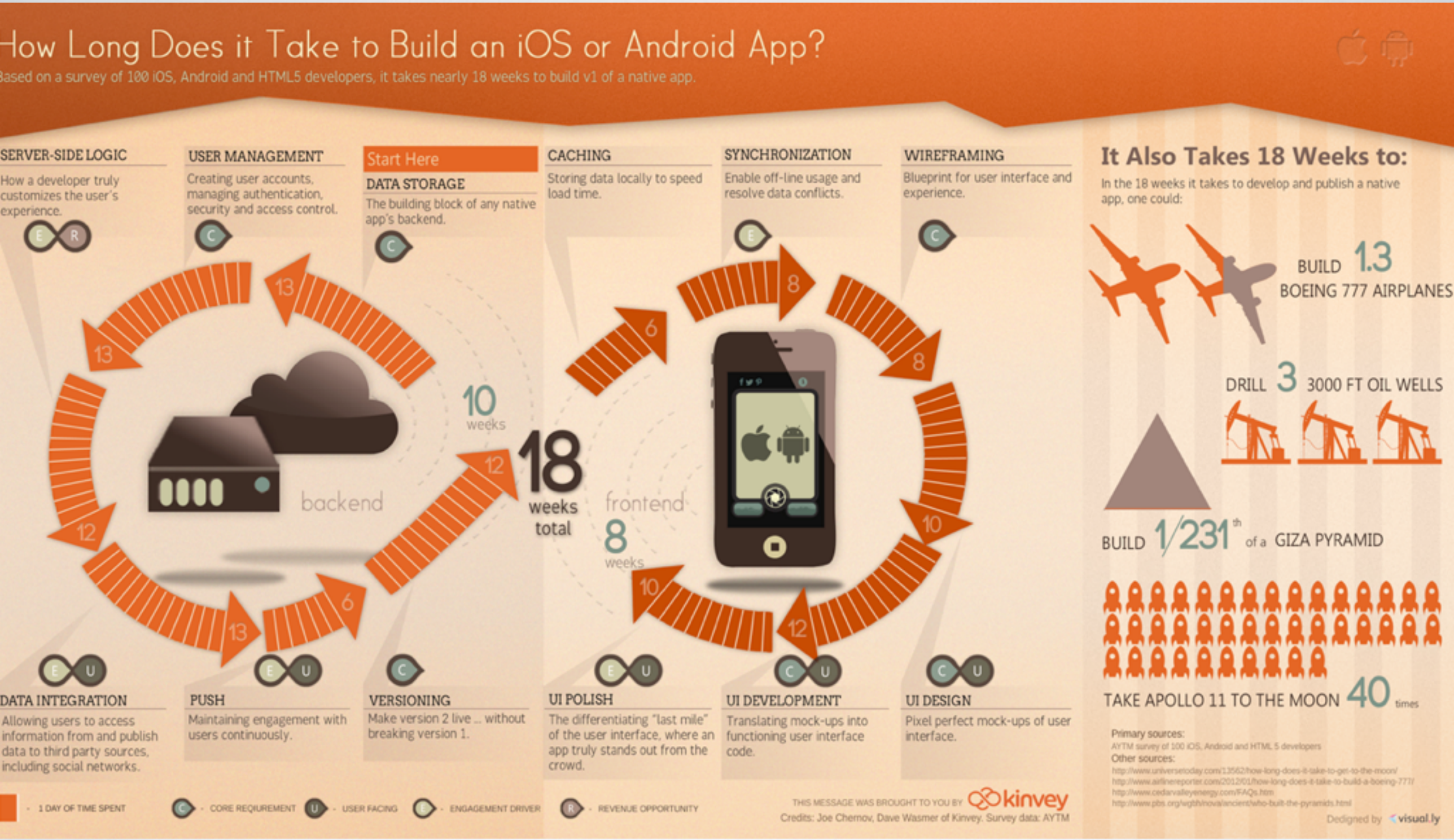


1.2 DESARROLLO DE APP

PRIMERAS PREGUNTAS

- ¿Cual es el propósito de mi app?
- ¿Dentro de que tipo de app encaja?
- ¿A que plataformas va dirigida?
- ¿Cual es mi mercado objetivo?
- ¿Que hace mi competencia?

Y DESPUÉS

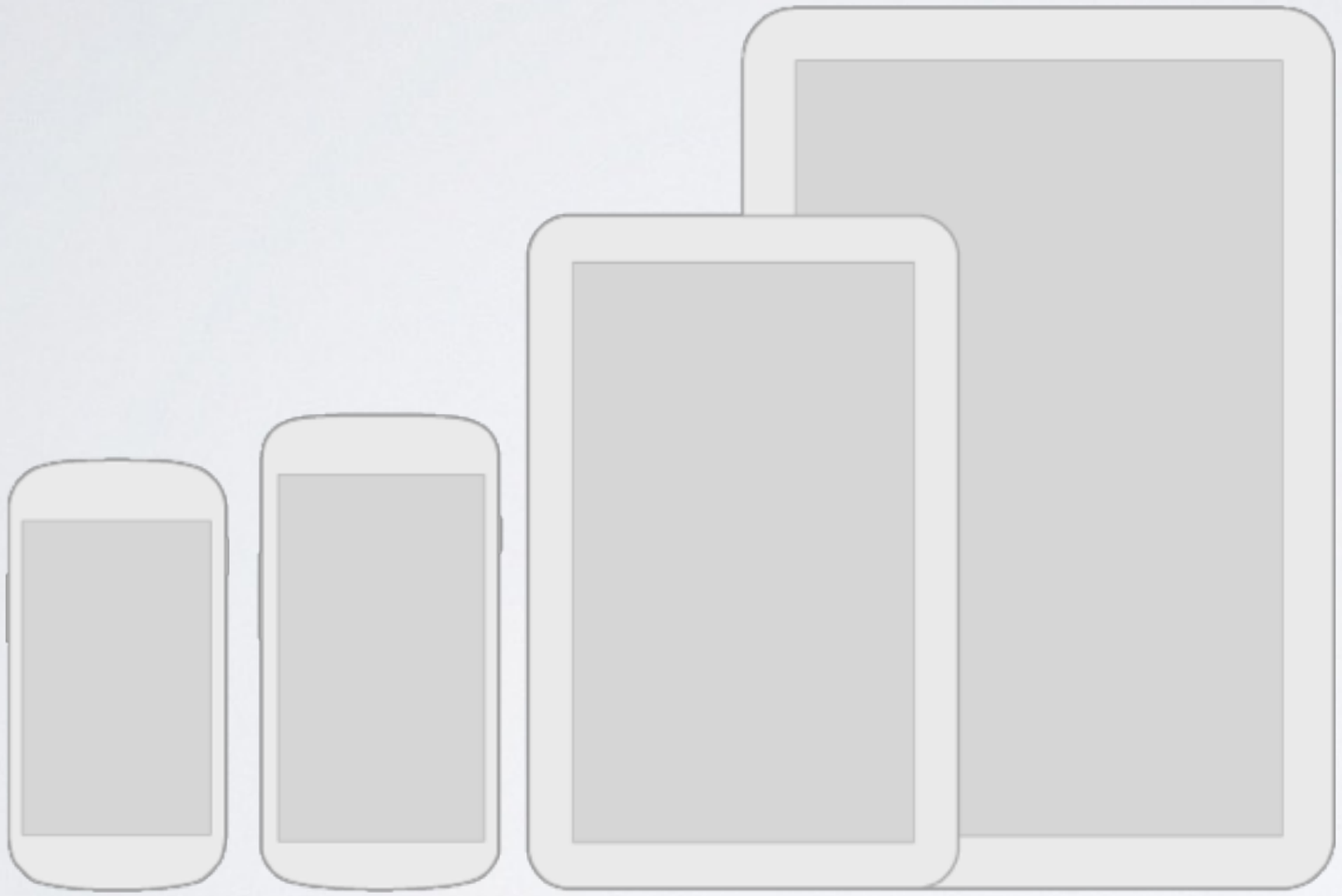


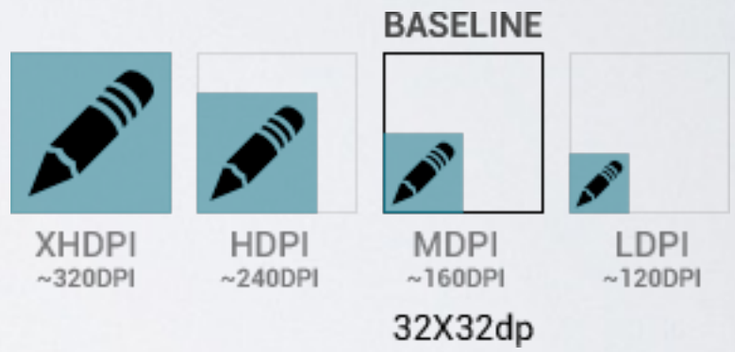
http://www.kinvey.com/blog/2086/how-long-does-it-take-to-build-a-mobile-app

SoftWhisper

Y CON LA UI

- Veremos esto en detalle poco a poco con los ejemplos y la estructura de una app. De entrada nos enfrentamos a esto:





1.3 HELLO WORLD

citic

SoftWhisper

APP BÁSICA

- Crearemos una app desde las plantillas del ADT
- Veremos la estructura de una App, que es cada componente esencial de la app y donde se encuentran.
- La recomendación, seguir el paradigma de convención sobre configuración.
- Eclipse > New Android Project ...

ACTIVITIES

- Componente esencial de las aplicaciones.
- Provee tanto a usuario como a desarrollador de un "tapiz" sobre el que pintar o tocar.
- Una activity nos provee de una ventana a la que agregar cualquier tipo de componente, desde un fragment, hasta un surface con la cámara de fotos.
- Durante su ciclo de vida podemos controlar los diferentes estadios de la aplicación y ejecutar las acciones que se requieran para ese momento.

PARTES DE LA APP

- Activities y sus View
- Servicios
- Provedores de contidos (/res)
- Intentes
- Broadcast receivers
- Widgets
- Notificationes (Toast por ejemplo)
- Pila de navegación

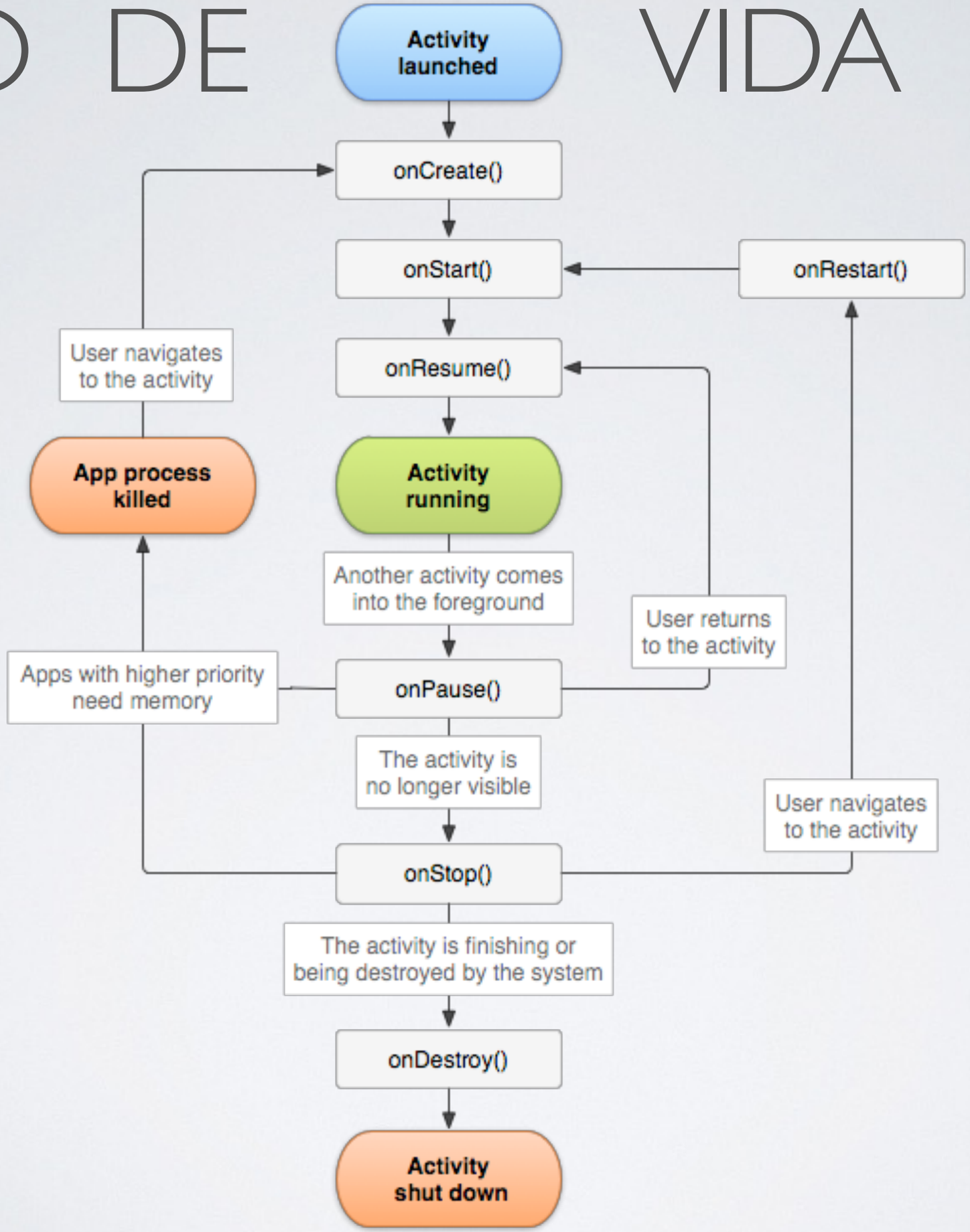
OTRAS PARTES DE LA APP

- Drawble
  - ldpi
  - mdpi
  - hdpi
  - xdpi
- Strings
- Animaciones
- Menús
- Recursos
- Assets
- Manifiesto

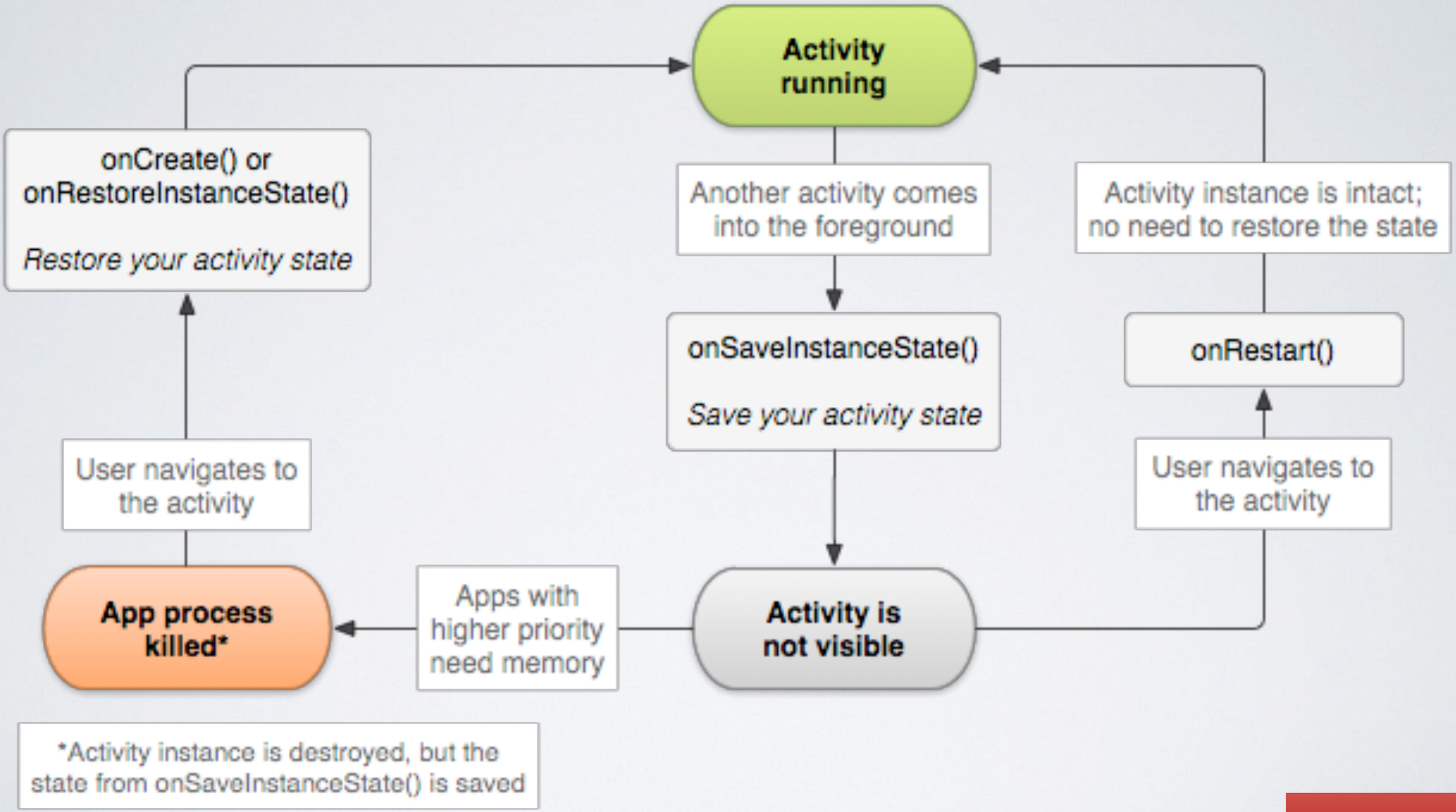
citic

SoftWhisper

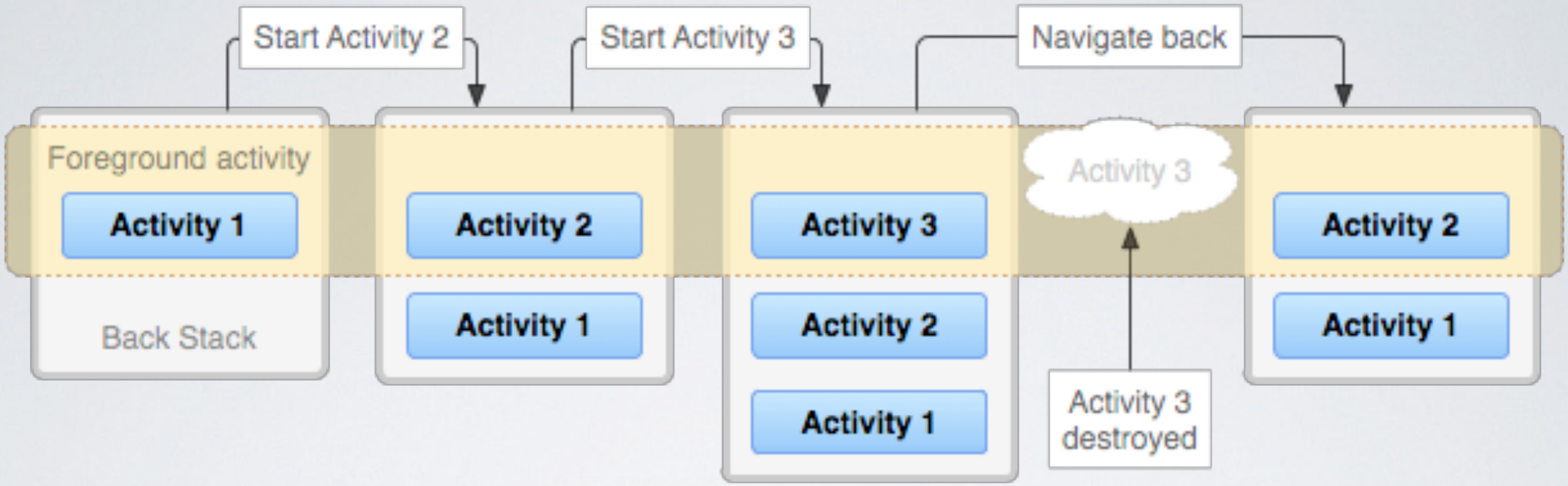
CICLO DE VIDA

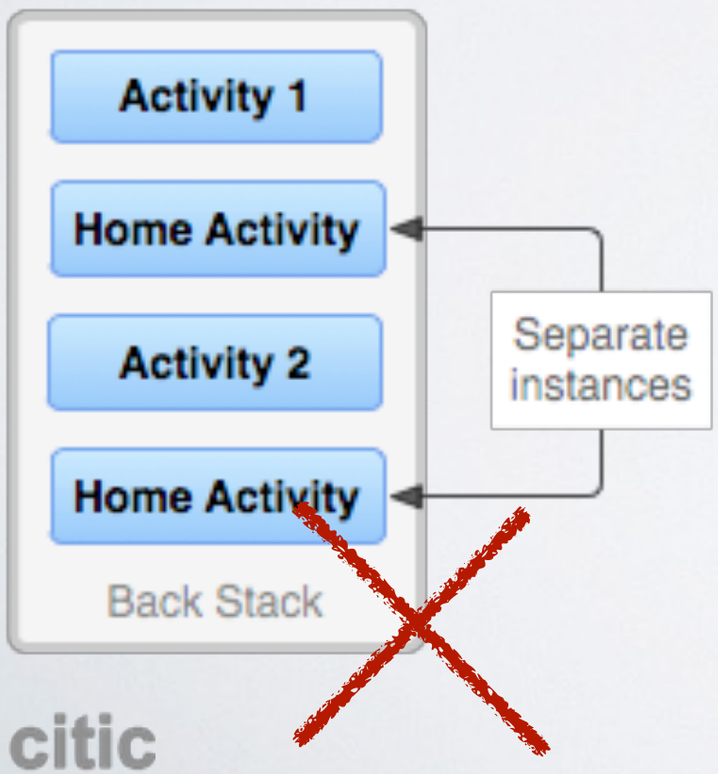


ACTIVITIES: PERSISTENCIA

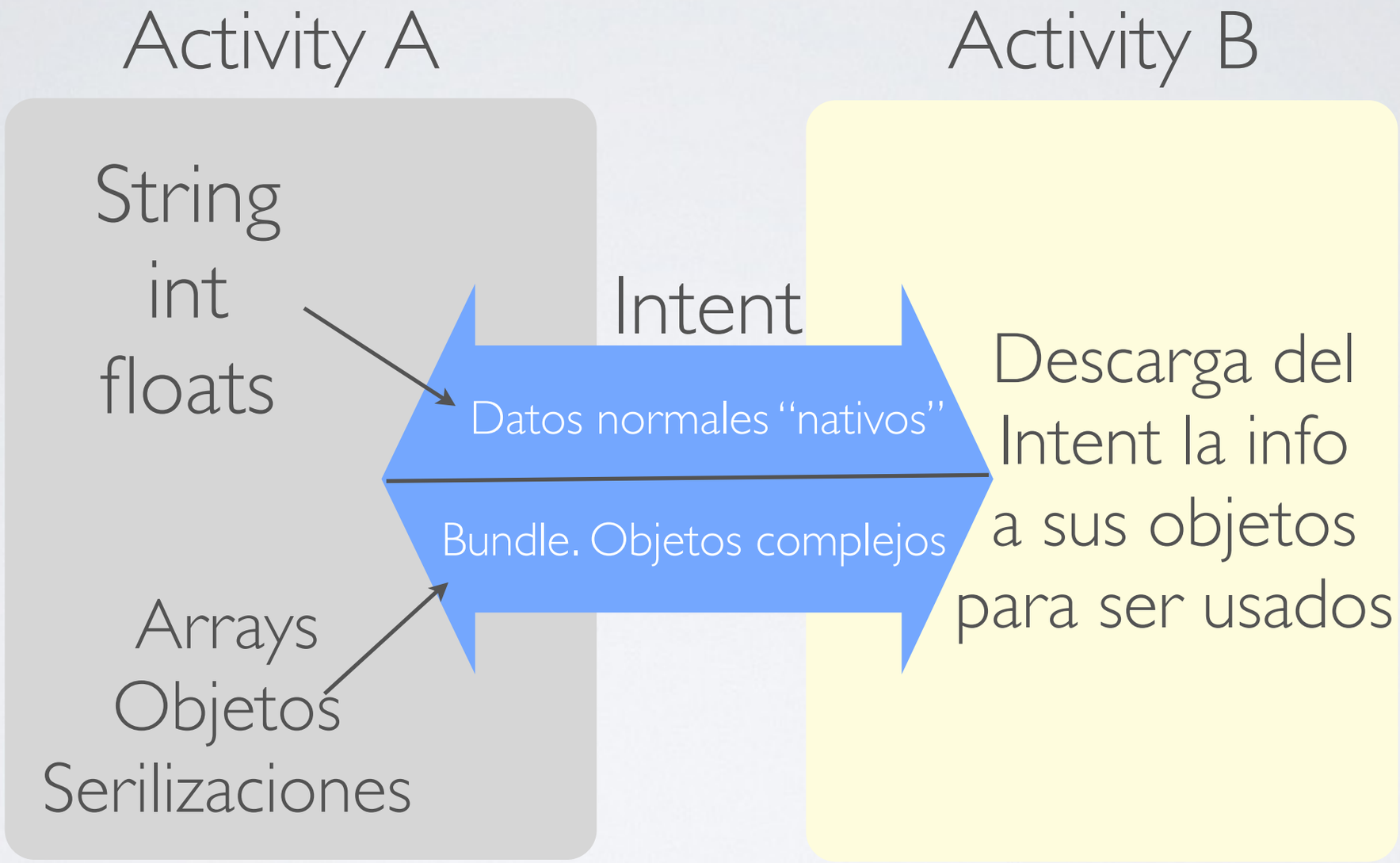


PILA DE NAVEGANCIÓN





NAVEGACIÓN POR INTENTS



APP ALUMNI DE EJEMPLO

- Esqueleto básico con los primeros pasos
- Objetivos
  - Conocer los componentes de la interfaz
  - Como anidar layouts para ordenar las UIs
  - Comprender el sistema de navegación
  - Picar las primeras líneas.

COMPONENTES DE ALUMNI

- Linear Layouts
- Buttons
- ViewText
- EditText
- Checks
- Selectores (Spinners)
- Mensajes Toast
- WebView

OBJETOS QUE VEREMOS

- Activity
- ArrayAdapters
- Listeners
- Intents
- Bundlers
- Selectores (Spinners)
- Mensajes Toast
- ListActivity
- pseudoFragments
- R
- Strings

MINIBREAK: GIT

- git clone <url> [destino]
- git add .
- git commit -m “Mensaje del commit”
- git remote add origin <url>
- git push <remoto> <rama>
- git pull <remoto> <rama>

2 INTENTS

Intents . IntentsFilters . ActivityResults .
BroadcastReceivers

citic

SoftWhisper

¿QUE ES UN INTENT?

- Es un "mensaje" capaz de activar
  - Activities
  - Servicios
  - Broadcast receivers
- No solo son una herramienta para la navegación.
- Comunicación entre aplicaciones.
- Uso de Intentes predefinidos en el Android

TIPOS DE INTENTS

- Explícitos
  - Donde el destinatario y la información es conocida
  - Su uso más frecuente es dentro de las propias aplicaciones
    - Lanzar activities o servicios
- Implícitos
  - El destinatario no es conocido
  - Usado para la comunicación con otras aplicaciones

2.1 INTENT FILTERS

- Describen las características de los componentes sobre los que se aplican
- Cada Intent puede contener entre 0 y n de estos filtros clasificados por
  - Acciones
  - Categorías
  - Datos (uris o tipos definidos)

EJEMPLOS FILTROS

- Acciones
  - EDIT
  - PICK
  - VIEW
  - READ
- Categorías
  - MAIN
  - LAUNCHER
  - DEFAULT
- DATA
  - scheme=”http”
  - mimeType=”audio/*”

citic

SoftWhisper

NOTEPAD

- Accede a las notas definidas por Google como vnd.google.notes
- Usa la acciones EDIT, PICK, VIEW para si manipulación
- Con INSERT le permite agregar nuevas notas
- En su activity principal hace uso de la URI lo datos devuelto y un CURSOR para leer los datos a través de un Adapter e inflar la lista

EJ: LECTOR DE HTML

- Con los filtros crearemos una app que se muestre como una opción más en el sistema a la hora de leer URI que sigan el schema http.

- Atiende a Intents implícitos

- GitHub: Proyecto CITIC03

INTENTS Y ACTIVITIES

- Configuración de los Intents de forma programática
- Usados para recibir información de vuelta de una activity
- Las acciones, categorías y schemas son los mismos que en la definición de los filters.

- Lanzamos las activities con startActivityForResult
  - arg0: el Intent
  - arg1: ID de la petición
- Callback a la vuelta: onActivityResult

EJEMPLO SELECTOR DE IMGS

- Desarrollaremos desde 0 un picker de la galería muy sencillo
- Usaremos la acción GET_CONTENT
- Como mimeType image/*
- La categoría adecuada es OPENABLE
- En la UI
  - Button
  - ImageView

2.2 BROADCAST RECEIVERS

¿QUE ES UN BR?

- Podemos entenderlo como un receptor de información de muchas de las cosas que pasan en el teléfono:
  - Envío de SMS
  - Llamadas recibidas
  - Estados de la batería
  - Conexiones bluethoot
  - ....

EJEMPLO DE BR

- Version reducida del TrojanCalls
  - Para el objetivo del curso no lleva la parte de control de SMSs ni similares (el código está disponible en muchas webs)

1. Al cargar la Activity registra el BR de fondo (transparante para el usuario)
2. La clase TrojanCall extiende de BR
3. Hace uso de las APIs de telefonía para acceder a los datos.

3 FRAGMENTS

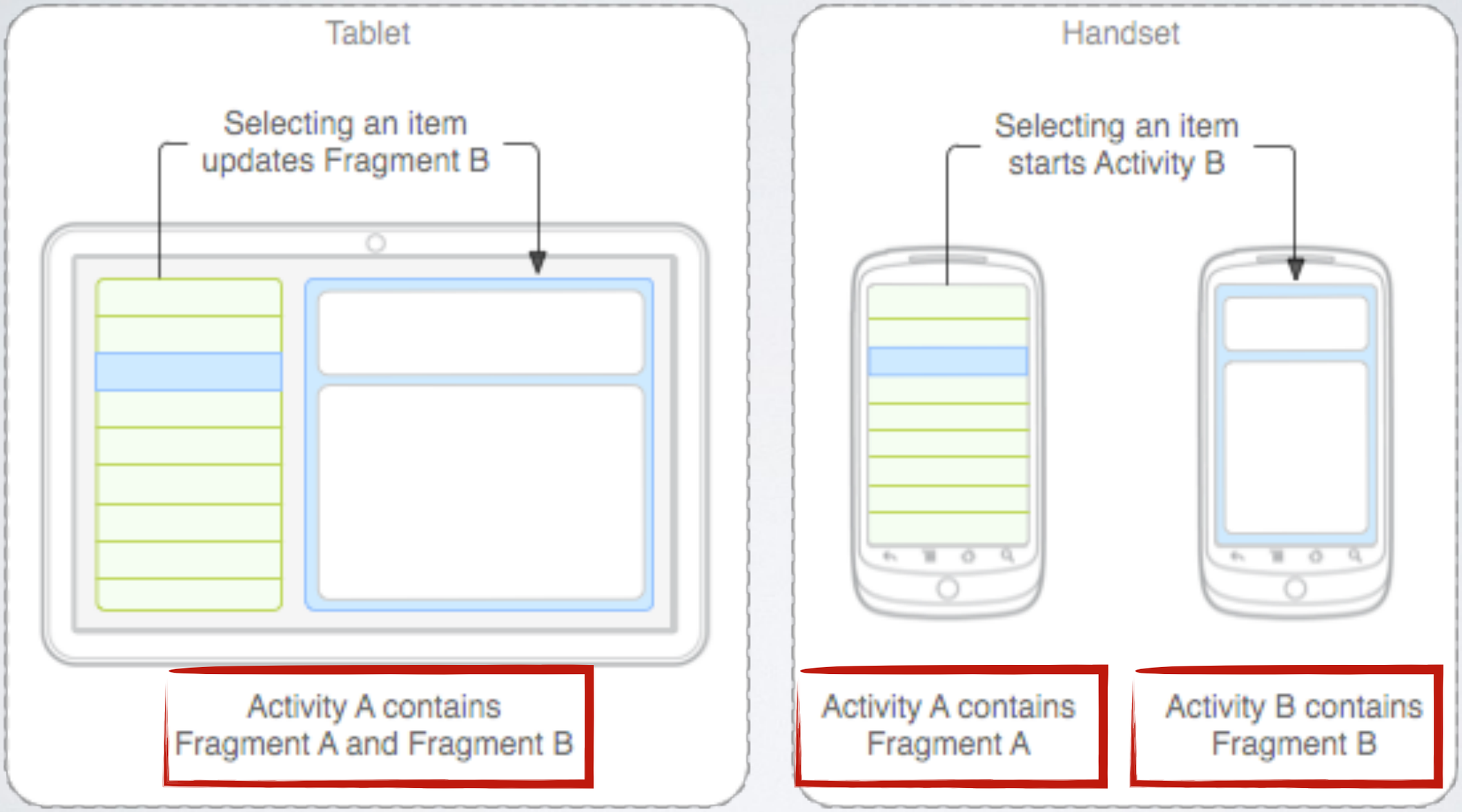
Fragments everywhere....

QUE ES UN FRAGMENT

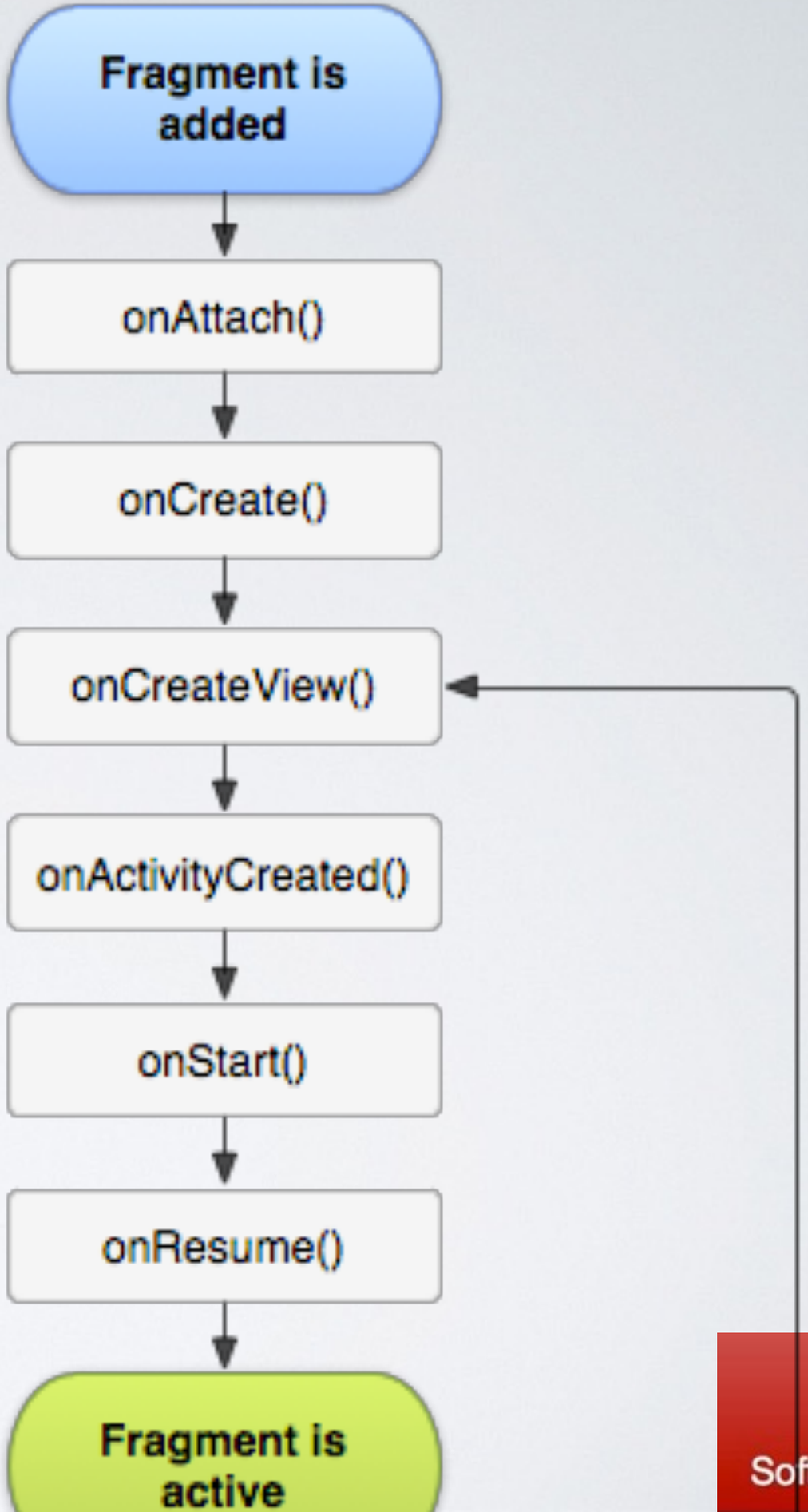
1. Podemos entender un fragment como una porción de la UI con comportamiento propio.

- Nos permite hacer diseños modulares
- Flexibles para varias pantallas de un forma especial
- Resusarlos a lo largo de varias activities
- Tienen su propio ciclo de vida
- Tipos predeterminados. También podemos crearlos de 0.

QUE ES UN FRAGMENT



CICLO DE VIDA



NATURALEZA

- Aparecen en API Level 11 por necesidades de diseño
- Nuevos componentes para las tablets
  - Necesidad de crear aplicaciones híbridas
  - Optimar los recursos en todos los dispositivos
- Compatibilidad inicial hasta Level 9
- Con la librería de dependencia podemos llegar hasta el Level 4

PRINCIPALES OBEJTOS

- FragmentActivity
  - Uno de los más importantes. De el heredan el FragmentList o el FragmentDialog
- FragmentManger que nos permite trabajar y acceder a estos.
- FragmentTransactions nos deja intercambian fragments sin recargar toda la vista. Gran avance en Android.
- Fragment ... el master. Si queremos crear nuestro propio fragment debemos extender de él.

USO

- Se pueden crear de forma programática a través del Fragmente Manager

- O podemos introducirlos en un layout de una Activity y acceder de forma similar a un recurso cualquiera.

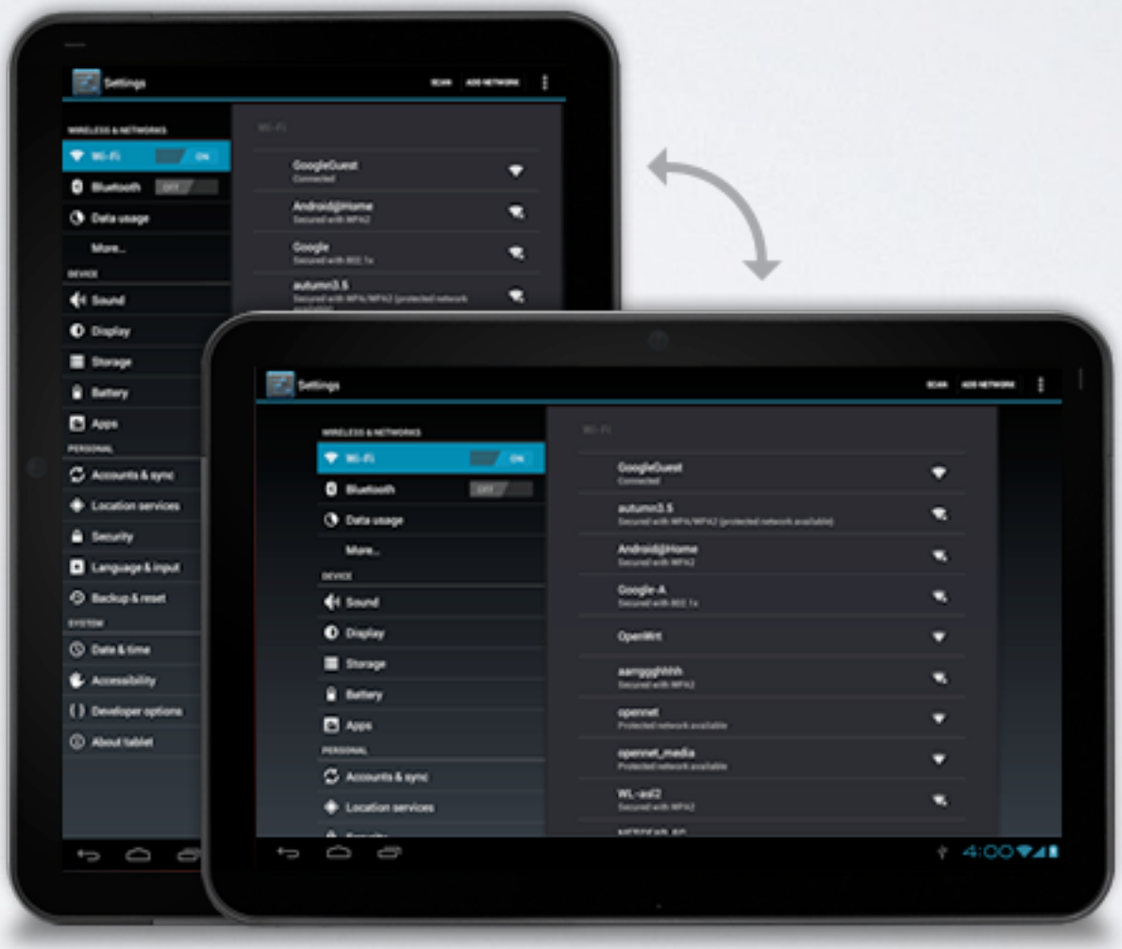
- Dentro de la UI, igual que las vistas los podemos rescatar una vez cargados con findFragmentByTag

DIALOG FRAGMENT

- El antiguo showDialog está deprecado
- Se muestra siempre flotante por encima de la interfaz

LIST FRAGMENTS

- Igual que las listas pero con nuestros propios adapters y vistas.
- Posibilidad de tener two-pane.



citic

SoftWhisper

4 CONEXIONES DE RED

HTTPURLCONNECTION

- Modelo clásico
- Envío de petición basado en URL
- Usamos StreamReaders para leer y parsear la respuesta

EJEMPLO COMPLETO

- App para ver los últimos terremotos en el mundo
- Usaremos el DOM Parser para acceder al objetos por el Path
- Envío de petición basado en URL
- Usamos StreamReaders para leer y parsear la respuesta con el objeto Document, su Builder y BuilderFactory.

4.1 AHORA EN SERIO

citic

SoftWhisper

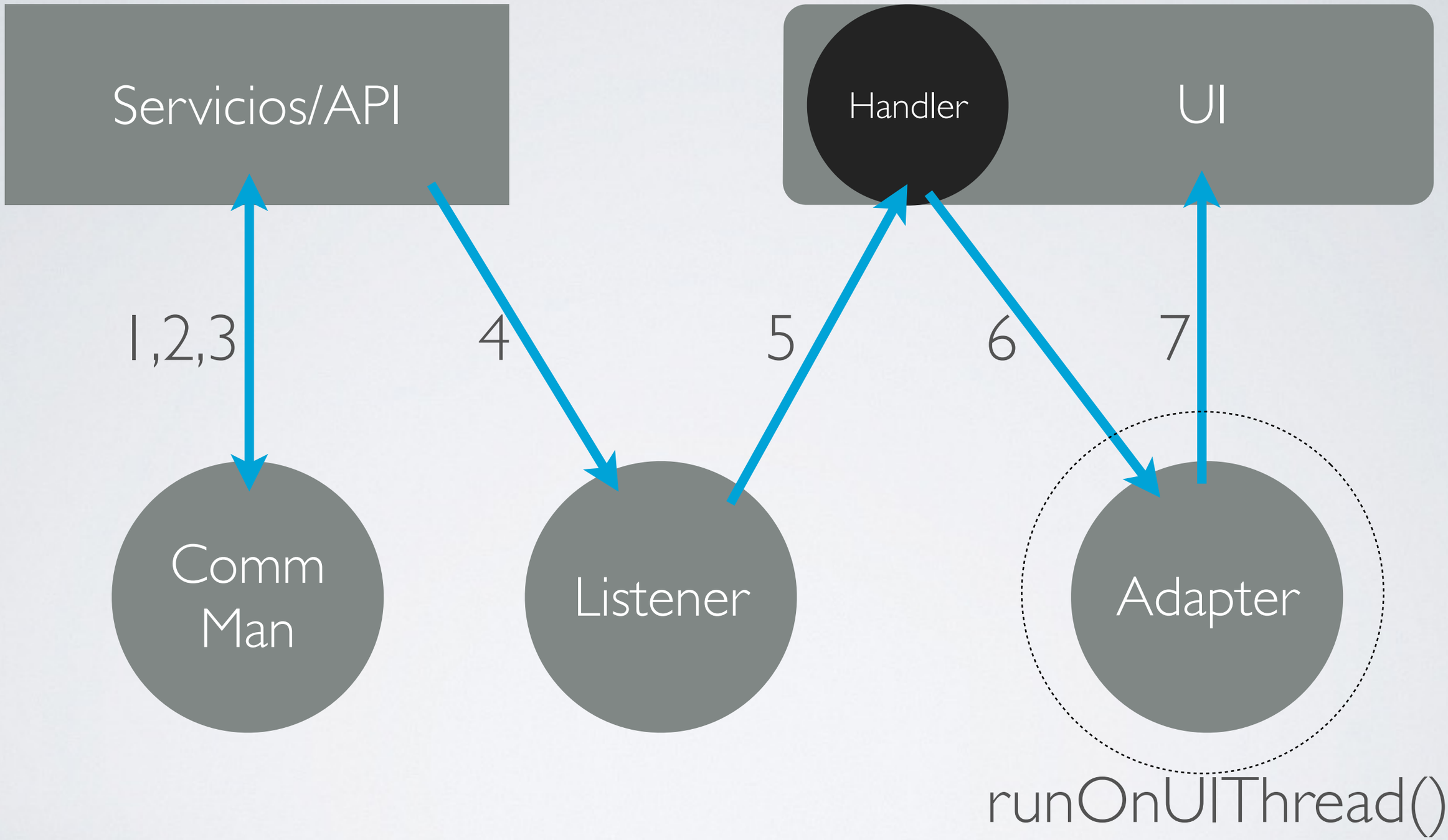
ESTRUCTURA DE RED/UI

- Siempre debemos buscar el menor coste de cambio
- Modularizar todo lo posible
- Establecer una serie de puntos centrales comunes:
  - BaseAdapter
  - CustomListener
  - CommManager

ACTIVITIES DESCARGADAS

- Eliminamos código funcional que no tiene que ver con la UI
- Delegamos la conexión en el CommManager
- Dejamos que el Listener se encargue de lo que sucede en la conexión.
- El Adapter prepara los contenidos.
- El Handler nos los vuelca a la UI.

CICLO



citic

SoftWhisper

4.2 REPASO SOBRE CÓDIGO

OBJETOS PARCELABLES

- Implementan la interfaz Parcelable
- A través del constructor se descodifica el objeto.
- En el writeParcel codificamos el objeto.
- Similar a la serialización pero con soporte de tipos de datos y sin necesidad de realizar casts.

4.3 LA CARA B :)

citic

SoftWhisper

LIBRERÍAS EXTERNAS

- Mantener aplicaciones con esta estructura conlleva costes elevados de desarrollo inicial.
- Existe un termino medio.
- Librerías como loopj AsyncHttp nos descargan de mucho de este trabajo.
- Nos limitamos a la recepción de datos y actualización de UI.

PEQUEÑO EJEMPLO

- Timeline de Twitter
- Descargamos el JSon ya en forma de objeto.
- Acceso directo a la propiedad que queremos.
- Recomendado un cliente (objeto por categoría de conexión) y otro por uso.
- Corren como AsyncTask y permiten acceso a la UI.

LOOPJ

- http://loopj.com/android-async-http/
- http://loopj.com/android-smart-image-view/

SoftWhisper

5 PERSISTENCIA I

SoftWhisper

MODELOS DE PERSISTENCIA

- SharedPreferences
- Estado de la aplicación e interfaz gráfica
- Ficheros
- BBDD (siguiente tema)

SISTEMA DE FICHEROS

- Memoria interna
- Tarjeta SD
- En los propios recursos de la aplicación

5.1 PERSISTENCIA BBDD

MODELOS DE PERSISTENCIA

- SQLite
  - Ligero
  - Simple
  - OpenSource
  - Estándar
- Content Providers
  - Compartir info
  - SMS, Agenda, etc...
  - Content_uri

SQLITE

- A nivel de NDK en C.
- Disponible en todas las versiones de Android
- SQLiteOpenHelper: CRUD
  - Cursor
    - Nativo de Android
    - Resultado de una query
    - Hace de iterador

CONTENT PROVIDERS

- Compartición de información entre aplicaciones
- Se separa la capa de datos de la aplicación
- Creados extendiendo ContentProvider
- Necesitan ser declarados en el Manifest
- Definen un URI propia y pública para el resto de aplicaciones.
- De implementar el CRUD compelto

CONTENT RESOLVERS

- Consumen información de los content providers
- Soporte para Cursor similar a SQLite

6 ACCESO AL HARDWARE

SENSORES DEL SISTEMA

- La SDK de Android incorpora un gran número de sensores
- La disponibilidad depende del dispositivo
  - Giroscopio
  - Orientación
  - Acelerómetro
  - Campo magnético

SENSORES DEL SISTEMA

- A partir del API Lvl 14 podemos encontrar también
  - Temperatura ambiente
  - Gravedad
  - Humedad relativa
  - Luz ambiente
  - ...

ACCESO A LOS SENSORES

- A través del SensorManager
- Obtenemos la instancia a tracés del servicio del sistema SENSOR_SERVICE
- Con el SensorEventListener implementamos los métodos necesarios para acceder a los cambios en el sensor.
- Recordar implementar el unregister en el onPause() de la activity.

LOS DATOS DEL SENSOR

- Debemos conocer e interpretar los valores que devuelve el sensor
- Ejemplo "Acelerómetro":
  - value[0] - eje x (lateral)
  - value[1] - eje y (longitudinal)
  - value[2] - eje z (vertical)
- Otro "Luz":
  - value[0] - intensidad de luz.

6.1 ESTACIÓN METEOROLÓGICA

citic

SoftWhisper

EJEMPLO ESTACIÓN

- Sensores
  - TYPE_LIGHT
  - TYPE_AMBIENT_TEMPERATURE
  - TYPE_PRESSURE
- Asociar la UI
- Instanciar el sensorManager
- Listener restantes
- Asociar los listeners
- Desregistrarlos

6.2 LA CÁMARA

ACCESO A LA CÁMARA

- Clase Camara como gestor del hardware
- SurfaceView para integrar la cámara con la UI
- Los SurfaceView se componen de un conjunto de callbacks similares a los del ciclo de vida de una Activity.
- En el ejemplo mandamos la imágenes de la cámara al Surface.

7 MAPAS

citic

SoftWhisper

USANDO MAPAS

- MapView
  - Componente que permite incrustar una mapa en la app.
  - Necesario usar en dev Google API.
  - Ahora ya la V2 :(

OBETENER API KEY V2

1. Instalar Google Play Services del SDK Manager
2. Obtener el SHA1 de la key que usemos (debuug)

*keytool -list -v -alias androiddebugkey *
*-keystore <path_to_debug_keystore>debug.keystore *
-storepass android -keypass android

3. https://code.google.com/apis/console
4. Accedemos a Servicios y Activamos GoogleMaps API v2 y GoogleMap for Android v2
5. En GM API v2 creamos una nueva clave para nuestra app

CAMBIOS SUSTANCIALES

- No es necesario usar Google APIs SDK Lvl X
- Importamos a nuestro workspace el proyeto de Google Services que marcamos como una librería.
- Configuramos el API Key en el Manifest con los persmisos necesarios (ej Maps1)
- Copiamos los ficheros proguard.cfg y proguard-project.txt a nuestro proyecto

CAMBIOS SUSTANCIALES

- El MapView se sustituye por un Fragment al que asociamos un objeto GoogleMap
- Animaciones y acciones de cámara integradas en el propio objeto.
- Simplificación del los overlays, sustituidos por nuevos Markers.
- En el Ejemplo Maps2 veremos muchas de las posibilidades.

7.1 LOCALIZACIÓN

SERVICIOS LOCALIZACIÓN

- Usamos el LocationManager:
  - Nos provee de los callbacks necesarios para trabajar con el servicio.
- LocationProviders
  - Diferentes tencologías que podemos usar para obtener la localización de un usuario.

SERVICIOS LOCALIZACIÓN

- Permisos necesarios:
  - ACCESS_FINE_LOCATION
  - ACCESS_COARSE_LOCATION

- Como provader podemos usar
  - GPS_PROVIDER
  - NETWORK_PROVIDER
  - PASSIVE_PROVIDER

BUENAS PRÁCTICAS

- Privacidad del usuario. Usar el servicio cuando es necesario.
- Muchas veces podemos acceder a la última posición conocida sin necesidad de atacar al GPS.
- Tener en cuanta el par +precisión/-batería
- El GPS puede ser lento y no funcionar en interiores. La espera hay que gestionarla.
- Comprobar la disponibilidad de los providers (pueden ser desactivados)

GEOCODING

- Tenemos soporte para
  - Reverse GC: calle a partir de lat/long
  - Forward GC: lat/long de una calle

8 SERVICIOS

citic

SoftWhisper

SERVICIOS

- Comoponentes que se ejecutan en segundo plano de forma transparente al usuario
- Es multitarea real, la app tiene sus propios porcesos
- Android ya implementa muchos de ellos con los Managers, por ejemplo el SensorManager
- Se definen a nivel de Manifest
- Extendemos la clase Service
  - implementamos el onStartCommand
  - se ejecutará al hacer un startService

BINDING DE SERVICIOS

- Asociación entre Activities y Serivces
  - referencia interna en la Activity al Servicio
- Un servicio puede estar en primer plano y pasar a un segundo plano si es necesario. El ejemplo más claro es el reproductor de música.

9 BLUETOOTH

citic

SoftWhisper

BLUETOOTH

- Protocolo de comunicación estándar
- Diseñado para determinadas situaciones
  - Corto alcance
  - Poca necesidad de ancho de banda
  - P2P

BLUETOOTH API

- Operaciones más comunes
  - Descubrimiento y conexión
  - Transferencia de datos corta

- Gestionado por el objeto BluetoothAdapter
- Conexiones a través de BluetoothServerSocket/ BluetoothSocket

9.1 NFC

citic

SoftWhisper

NFC

- La comunicación se produce en distancias cortas
  - Dispositivo <-> Dispositivo
  - Dispositivo <- TAG
- Usa el estándar NDEF para el envío de mensajes

NFC API

- API LVL 9+
  - Necesita librería de dependencia para algunos métodos
- Gestión a través del NfcAdapter
- Ndef* define lo relacionado con la creación de TAGS.
- Define en un IntentFilter el tipo de mensajes que va a recibir.
- Ampliar con Android Beam Messages

10 PROCESSING

citic

SoftWhisper

PROCESSING

- http://processing.org/
- Entorno para prototipado rápido
- Define un DSL sobre la API de Android
- Soporta el 90% de las funcionales de Android
- Dispone de emulador y ejecución directa en dispositivo

PROCESSING

```
void draw() {
  ellipse(mouseX, mouseY,
          mouseY - mouseX, mouseX - mouseY);
}
```

- Pocas líneas de código
- Exportable a un apk compilable.
- Compatibilidad con HTML5

DISEÑO

citic

SoftWhisper

TENDENCIAS

- 9 *Patchea* todas las imágenes que puedas
- Paradigma direccional
  - Vertical: Mismo contenido
  - Horizontal: Contenidos diferente pero relacionado
- PagerView componente más usado
- Fragments para todos