Bootcamp: Desarrollo en Android

Medellín, Febrero 2015

#cappacitate2015 | @appsco | @Ministerio_Tic | @Cymetria | @unmultimedio

Agenda

TEMA	SESIÓN	TEMA	SESIÓN
Historia	1	Geolocalización	2
Workflow	1	IDE/Debug	3
“Hello World”	1	Fragmentos	3
Ciclo de vida	1	Interfaces	3
Action Bar	1	Notificaciones	4
XML Layouts	1	Consumo APIs	4
Menús	2	GCM	4
Persistencia	2	Q&A	1-4

Historia: Android como OS

- Android Inc. (2005 Google)
- +1MM diarios
- Open source
- C | C++ | Java
- Construido y compilado sobre Linux (apps como usuarios)
- DB relacional (SQLite)
- OpenGL (v2.0, v3.0)
- Mercado móvil 2007 (iPhone)
- C2DM (Push)
- WebKit (Chromium)
- GPS, acelerómetro, giroscopio, proximidad, luz.
- Emulador sobre Eclipse*
- Multitarea real desde GB

*Genymotion!

Historia: Versiones

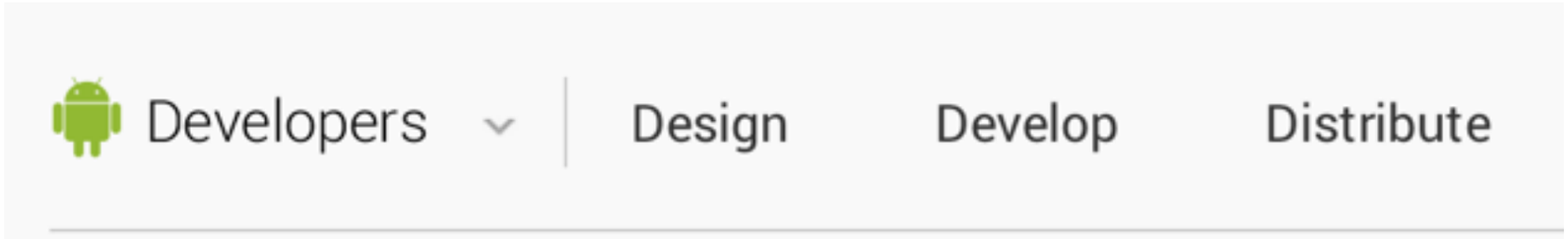
Historia: Versiones

POSTRES	
Apple Pie	Gingerbread
Banana Bread	Honeycomb
Cupcake	Ice Cream Sandwich
Donut	Jelly Bean
Eclair	Kit Kat
Froyo	Lollipop

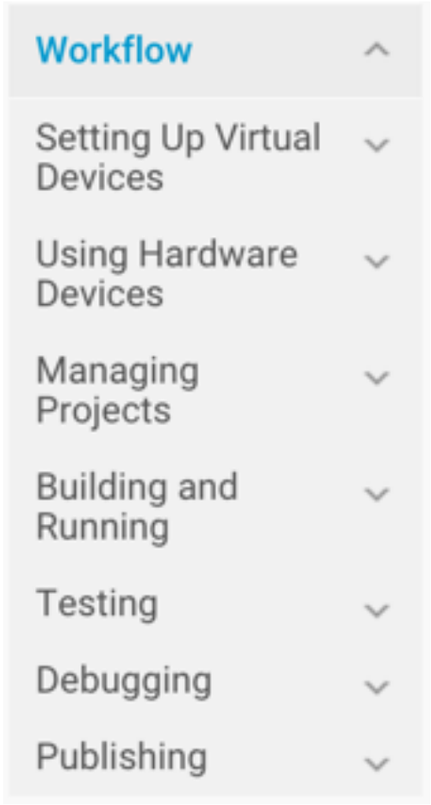
Historia: Versiones

NOMBRE VERSIÓN	API	NOMBRE VERSIÓN	API
Apple Pie	1	Gingerbread	9-10
Banana Bread	2	Honeycomb	11-13
Cupcake	3	Ice Cream Sandwich	14-15
Donut	4	Jelly Bean	16-18
Eclair	5-7	Kit Kat	19-20*
Froyo	8	Lollipop	21

Workflow: DDD



- IDE
- Dispositivos/Emuladores
- Test App
- Code!!!
- Desplegar
- Depuración
- Firmar, test a producción
- Alfa, beta (privadas, públicas)
- Publicar, promocionar.

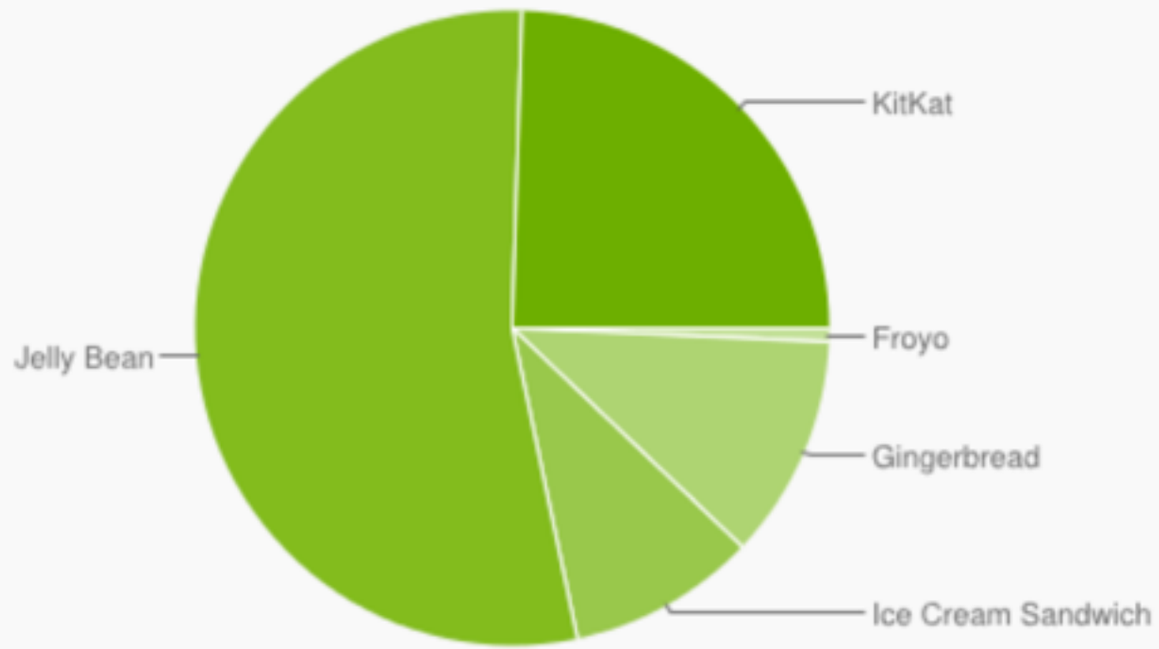


Workflow: Soporte

Dashboard 12-2014

Versiones

Version	Codename	API	Distribution
2.2	Froyo	8	0.7%
2.3.3 - 2.3.7	Gingerbread	10	11.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	9.6%
4.1.x	Jelly Bean	16	25.1%
4.2.x		17	20.7%
4.3		18	8.0%
4.4	KitKat	19	24.5%



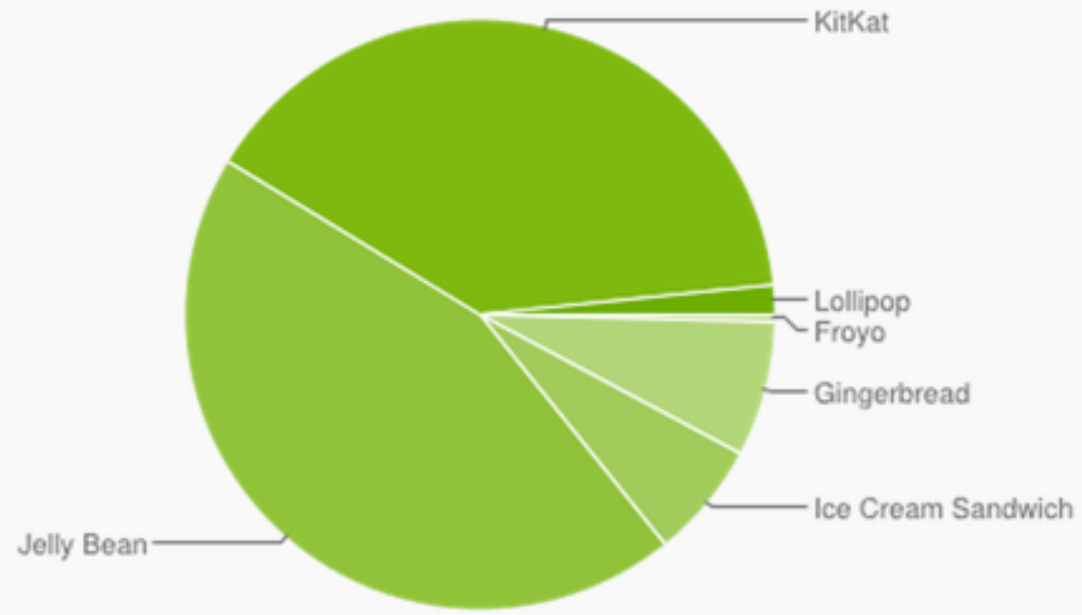
Densidades y tamaños de pantalla
Versión de OpenGL

Workflow: Soporte

Dashboard 02-2015

Versiones

Version	Codename	API	Distribution
2.2	Froyo	8	0.4%
2.3.3 - 2.3.7	Gingerbread	10	7.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	6.4%
4.1.x	Jelly Bean	16	18.4%
4.2.x		17	19.8%
4.3		18	6.3%
4.4	KitKat	19	39.7%
5.0	Lollipop	21	1.6%



Densidades y tamaños de pantalla

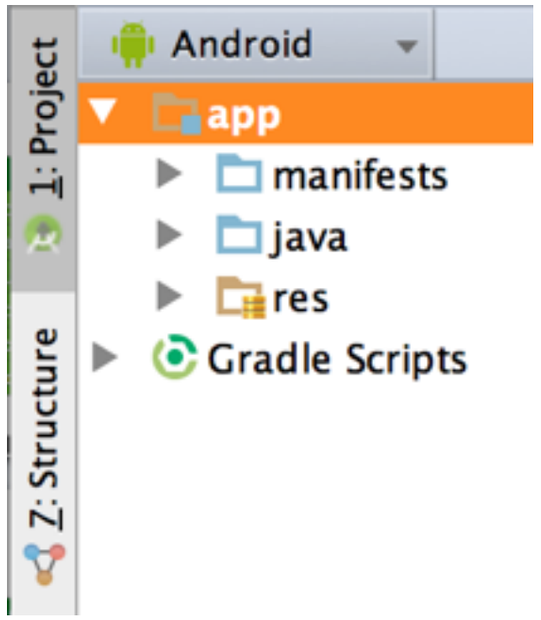
Versión de OpenGL

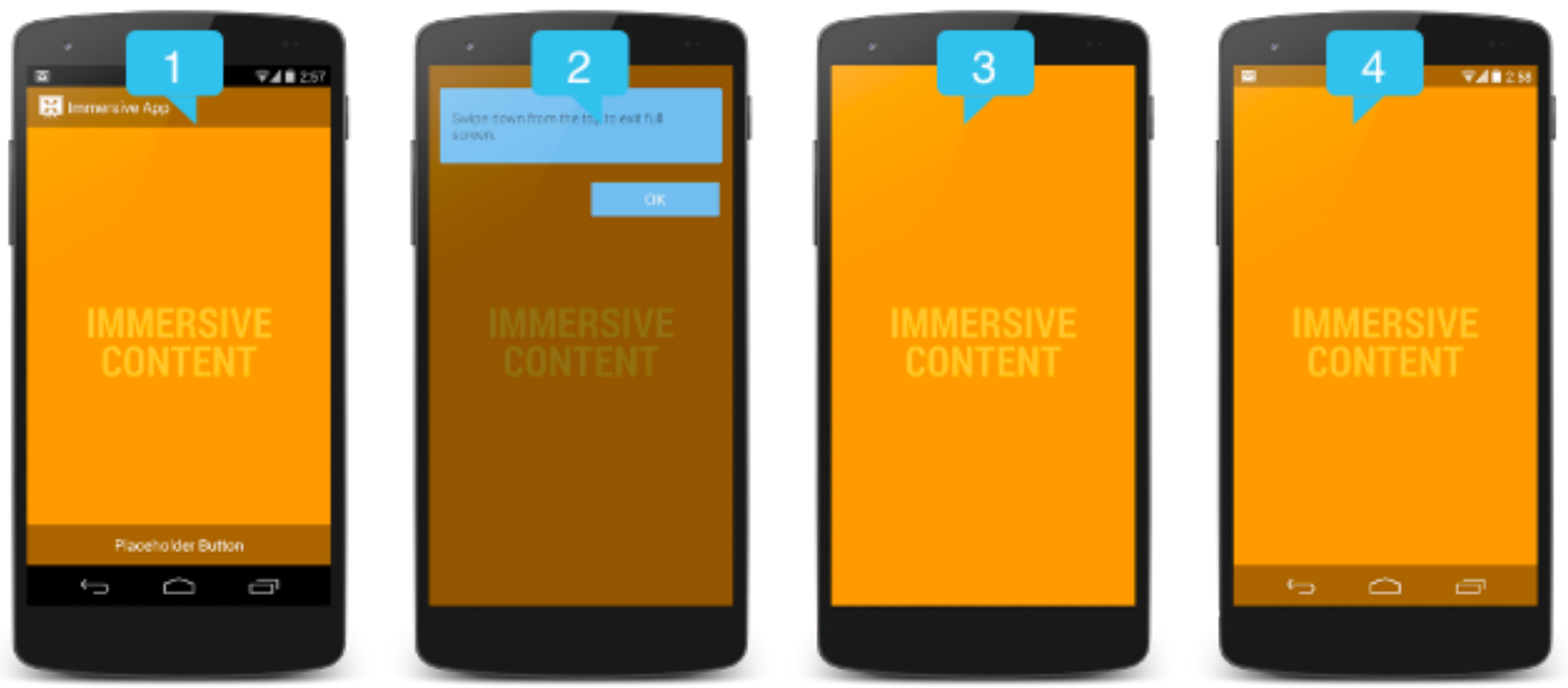
Workflow: Requisitos

- Android Studio (Beta estable)
- Eclipse con ADT oficial (opcional)
- Acceso a la documentación y referencia (developer.android.com)
- Equipo Android de Depuración (opcional)
  - Windows OEM Driver
  - Linux “/etc/udev/rules.d/51-android.rules” file
  - MacOSX :)

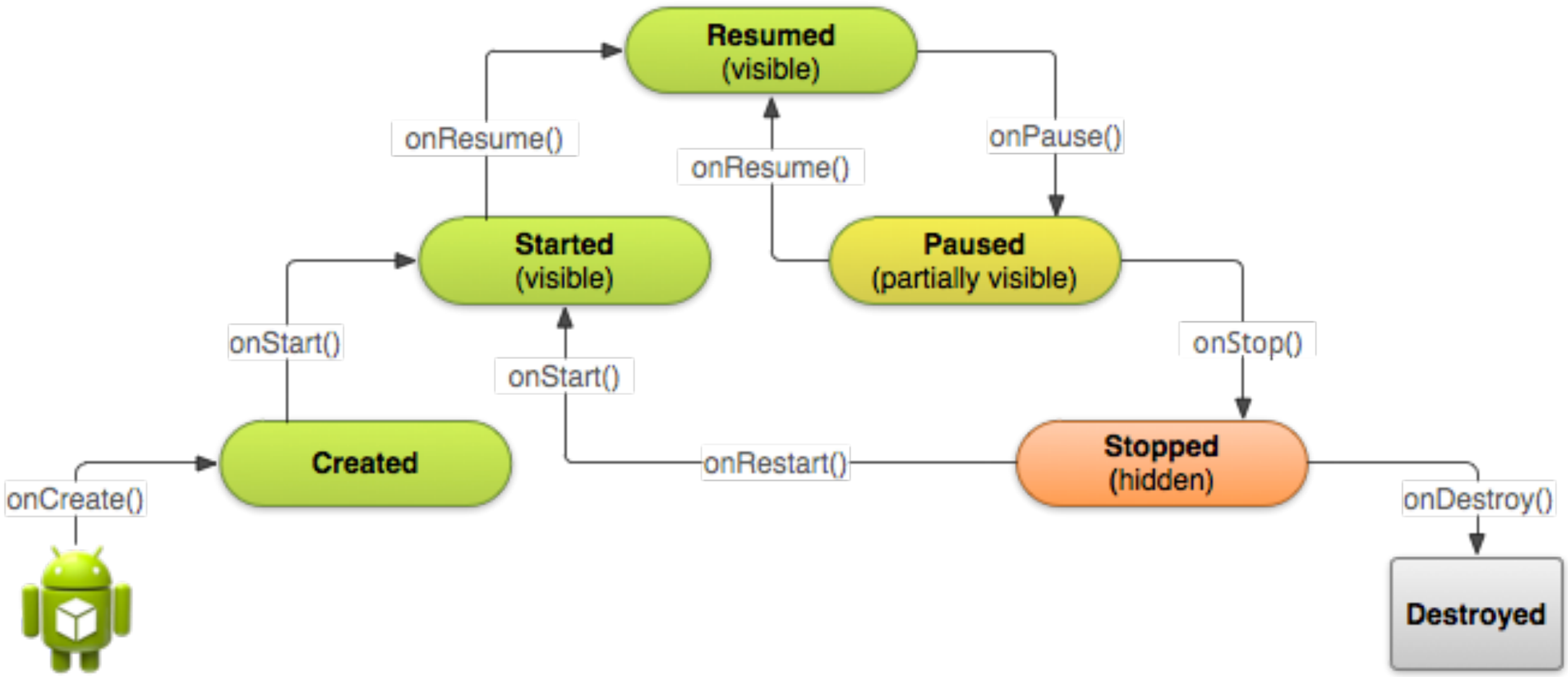
“Hola Mundo”

- Componentes de un proyecto
- Actividades
- Recursos
- Manifiesto
  - Layouts
  - Strings
- Toast
- Intent



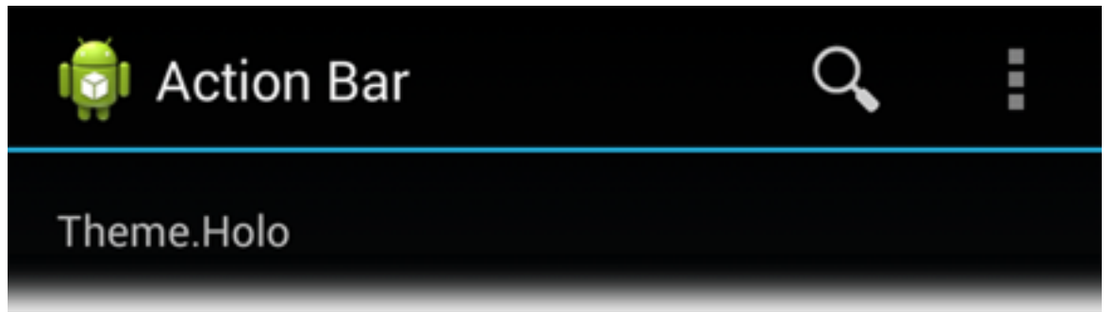


Ciclo de vida



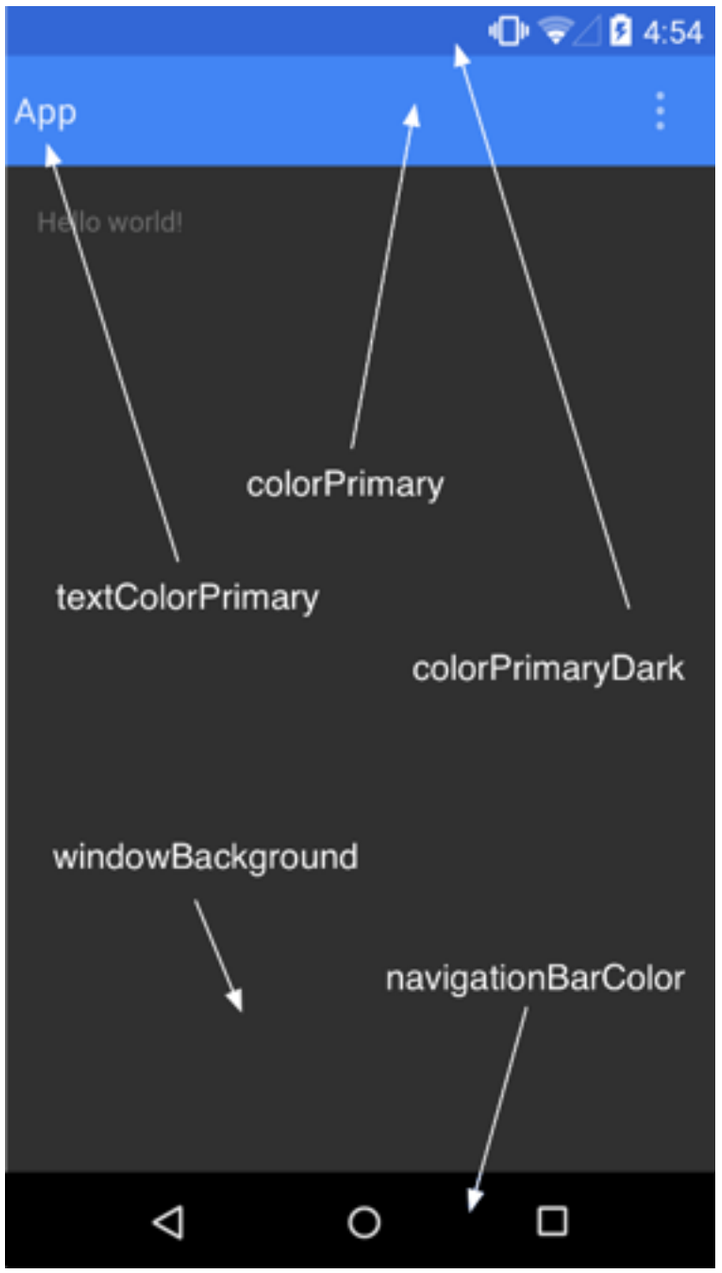
Action Bar

- Versión de soporte? AppCompat
- Menús
- Iconos como acciones
- Estilos
- Recursos



Action Bar

- Versión de soporte? AppCompat
- Menús
- Iconos como acciones
- Estilos
- Recursos



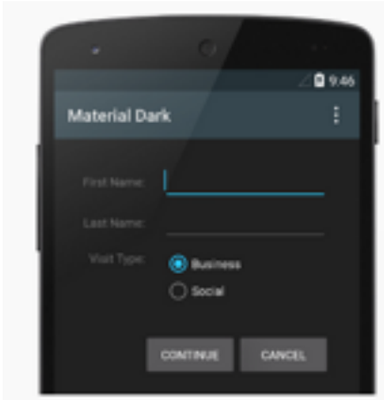


Figure 1. Dark material theme

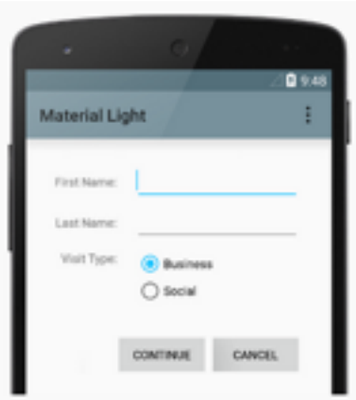
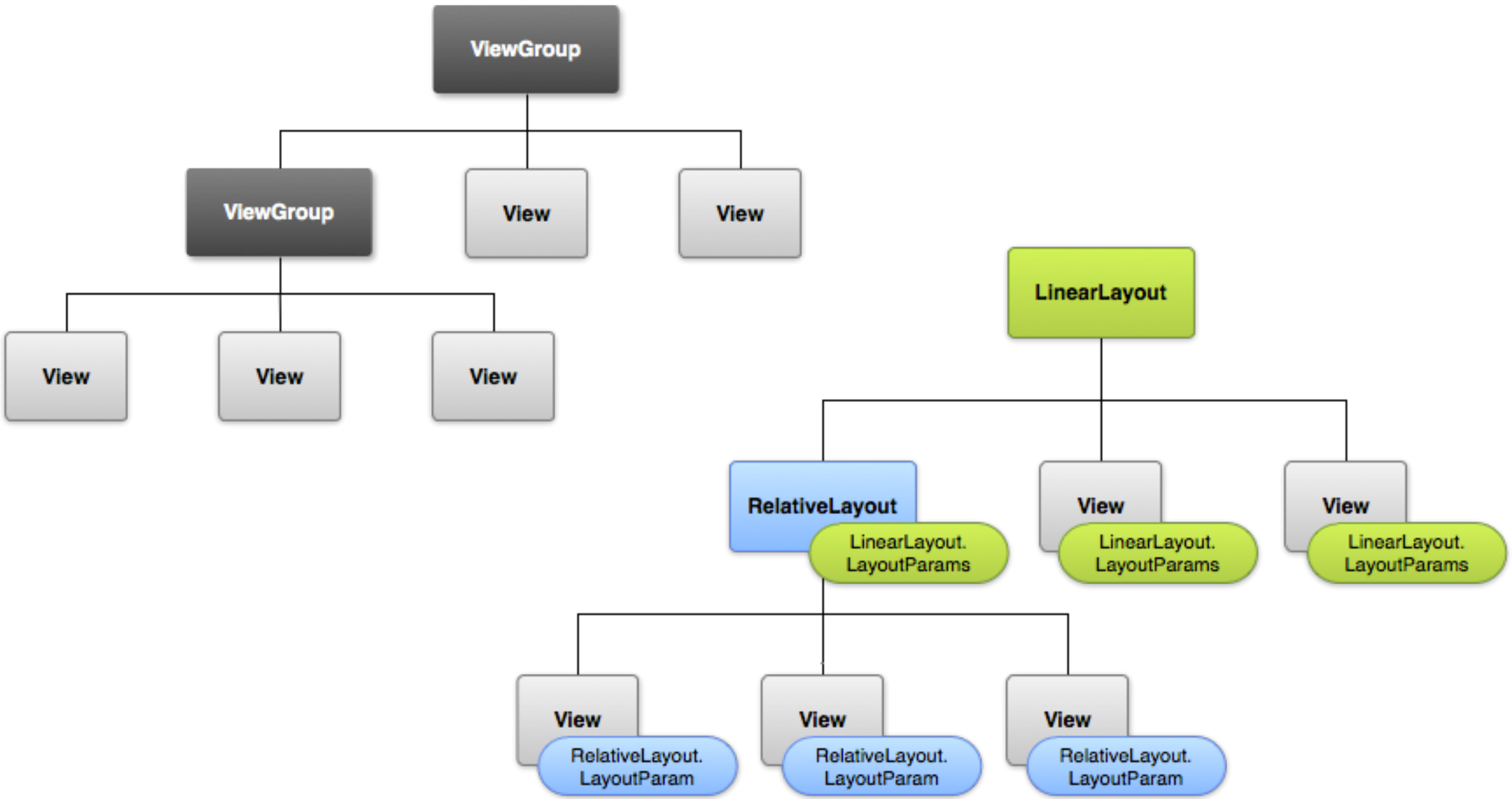


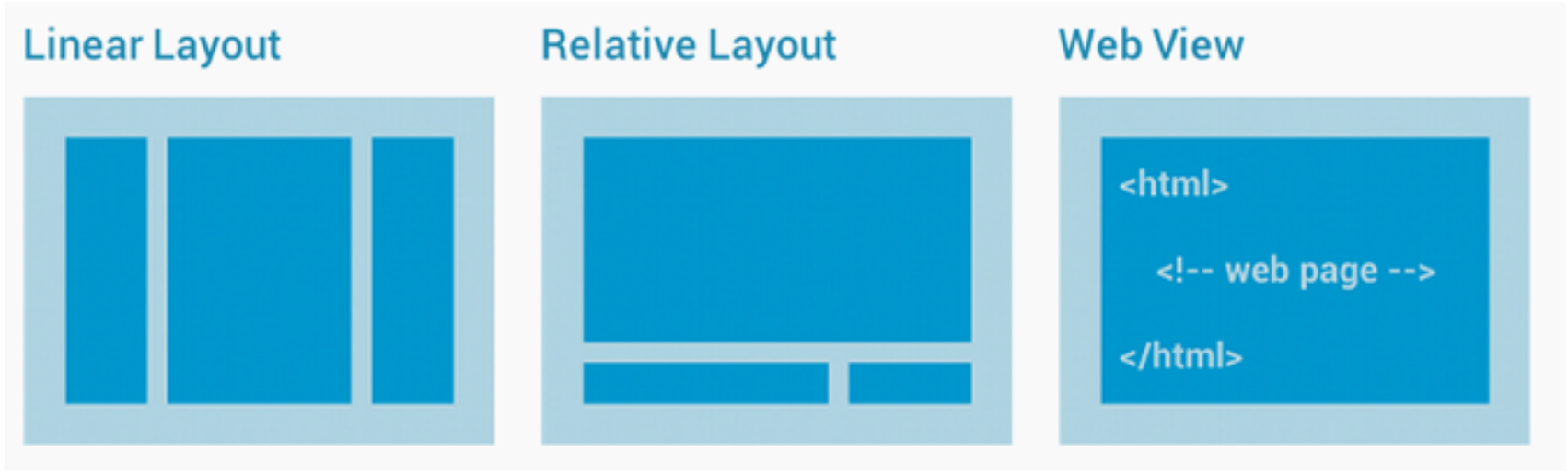
Figure 2. Light material theme

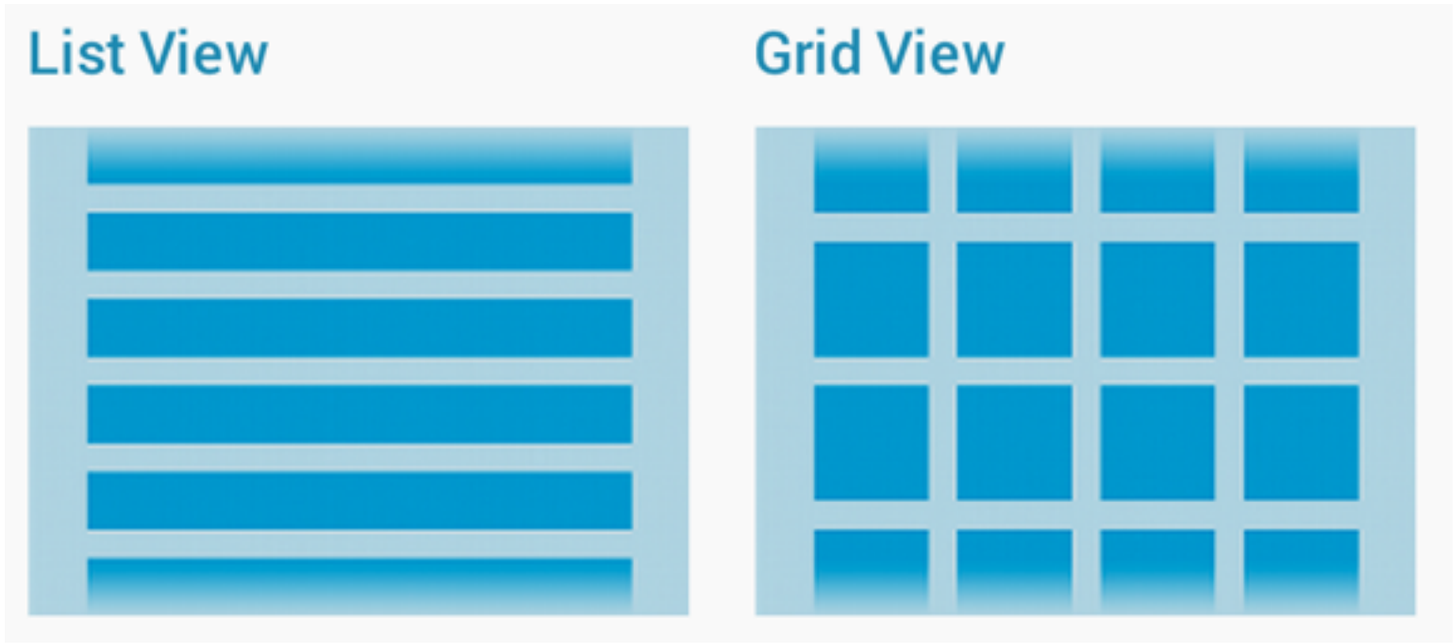
Layouts: Vistas



Layouts: Organizaciones Comunes

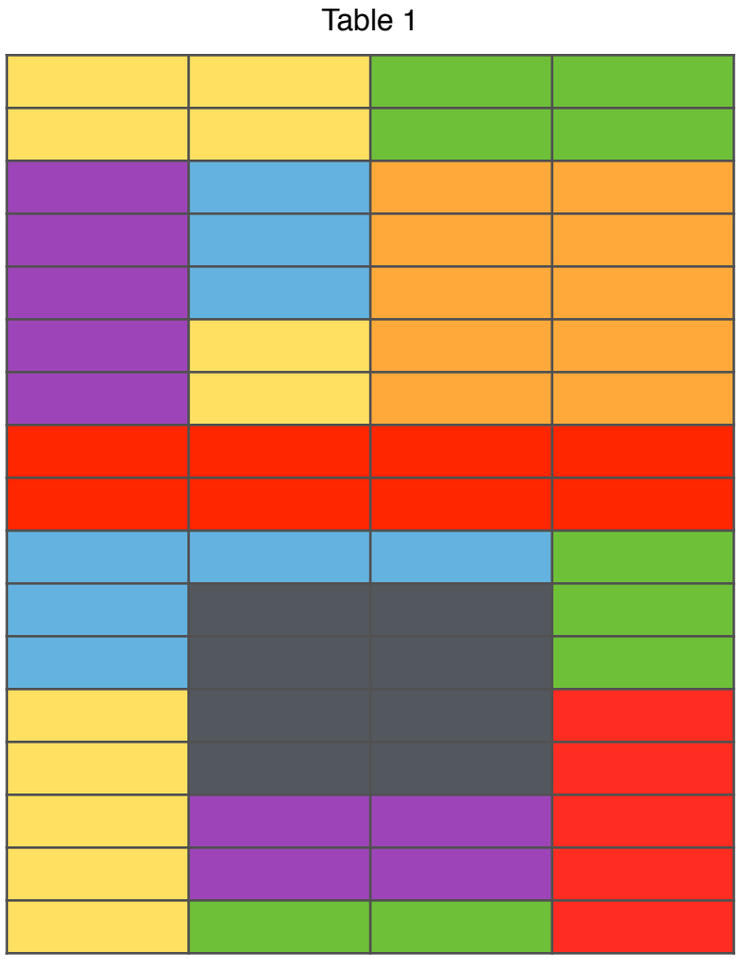
- Linear
- Relative
- WebView
- ListView
- GridView

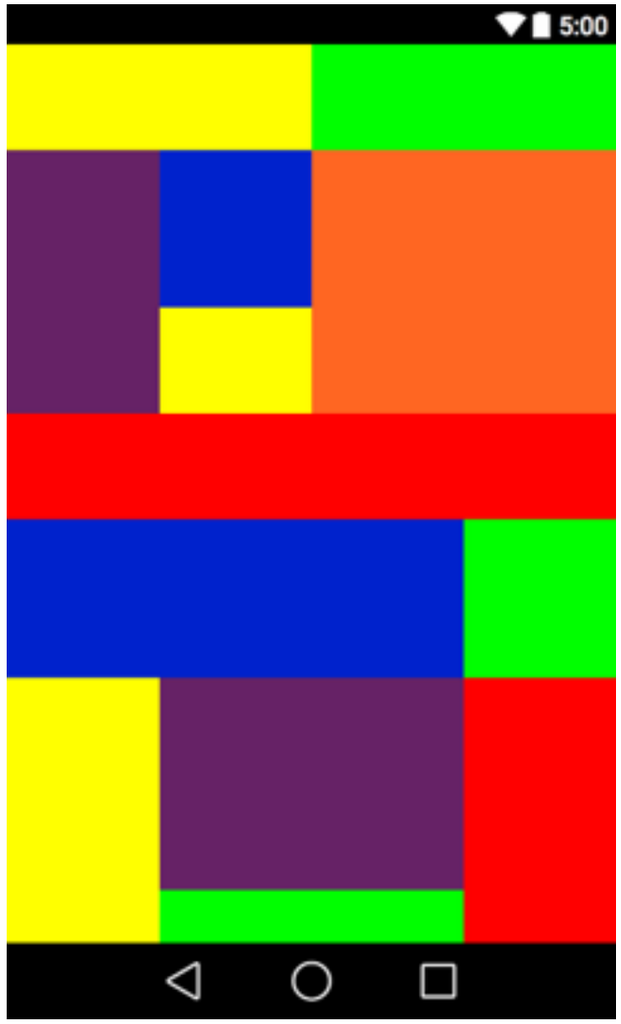




Layouts: Pesos

- Linear Layout
- No hay medidas estáicas
- Orientaciones y pesos
- ¿Organización?





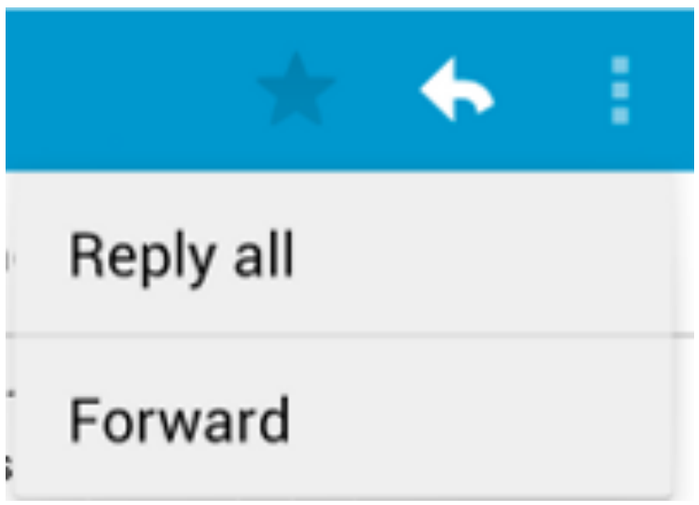
Agenda

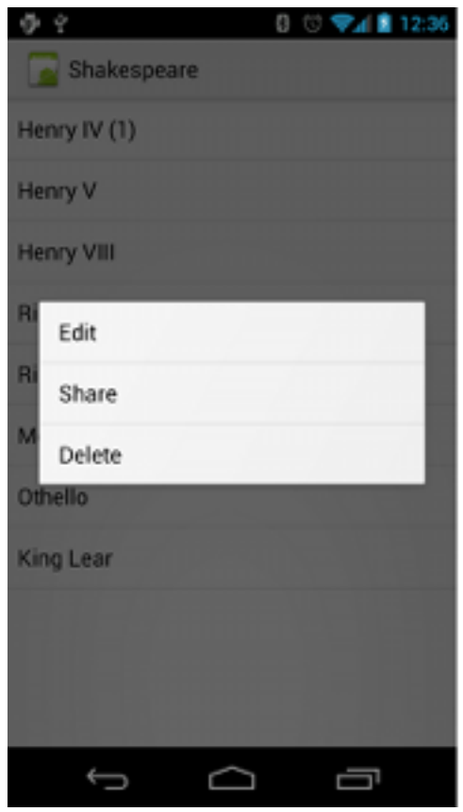
TEMA	SESIÓN	TEMA	SESIÓN
Historia	1	Geolocalización	2
Workflow	1	IDE/Debug	3
“Hello World”	1	Fragmentos	3
Ciclo de vida	1	Interfaces	3
Action Bar	1	Notificaciones	4
XML Layouts	1	Consumo APIs	4
Menús	2	GCM	4
Persistencia	2	Q&A	1-4

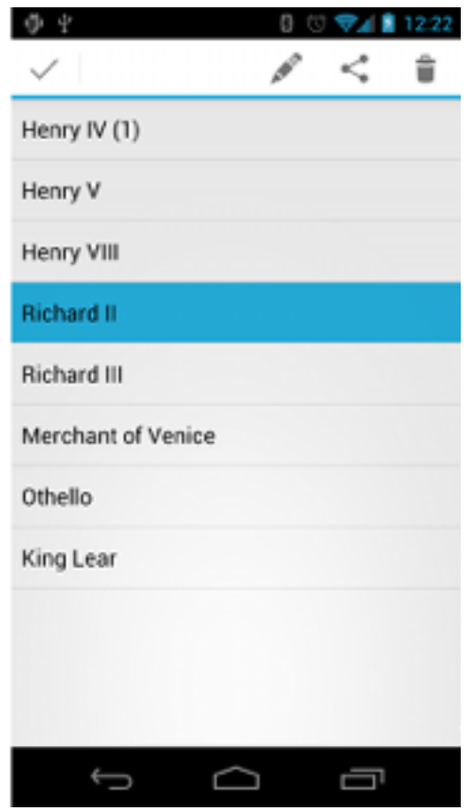
Menús

- Opciones (Acciones) del ActionBar

- Menú tipo “pop-up”

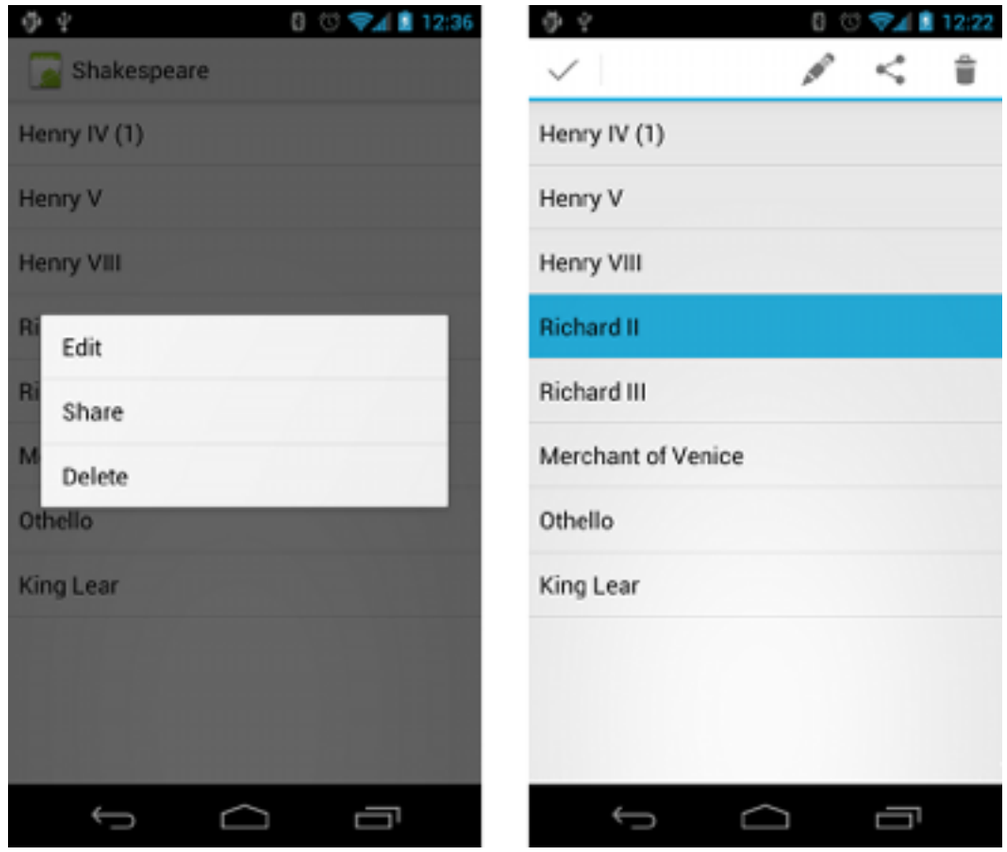






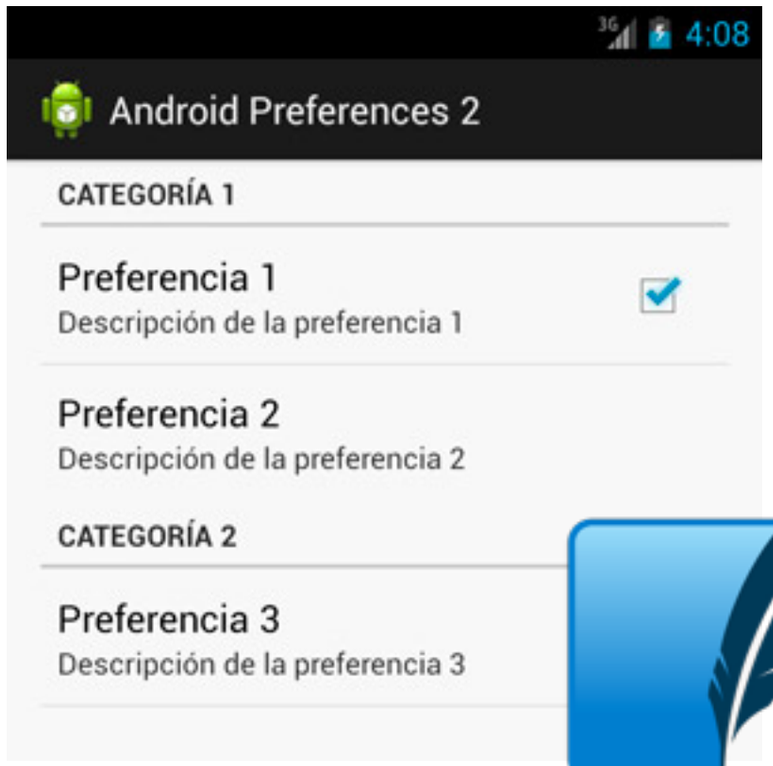
Menús Contextuales

- Fragmentos de listas
- Envío de información entre fragmentos
- Adaptadores personalizados
- Control de eventos de items
- Menús contextuales por ítem
- Actualización de listas



Persistencia

- Preferencias compartidas
- SQLite
- Archivos



Persistencia: SQLite

- Paquete contenedor: android.database.sqlite
- Clase contrato
- Constantes de nombres de tablas, vistas, columnas, tipos de datos
- Con un contrato se puede replicar las tablas, consultas y ejecuciones en todo el código

Persistencia: SQLite

```
public class DatabaseContract {
    //Tabla Usuarios
    public static class Users implements BaseColumns{
        public static final String TABLE_NAME = "users";
        public static final String COLUMN_NAME_NAME = "name";
        public static final String COLUMN_NAME_DRINK = "drink";
        public static final String COLUMN_NAME_SPORT = "sport";
    }
    // Otras tablas, vistas...
}
```

Persistencia: SQLite

```
public class Ayudante extends SQLiteOpenHelper {

    public static final String DATABASE_NAME = "nombrebase.db";
    public static final int DATABASE_VERSION = 1;
    public static final String SQL_CREATE_USERS =
        "CREATE TABLE " + Users.TABLE_NAME
        + " ("+ Users._ID + " INTEGER PRIMARY KEY, "
        + Users.COLUMN_NAME_NAME + " TEXT, "
        + Users.COLUMN_NAME_DRINK + " TEXT, "
        + Users.COLUMN_NAME_SPORT + " TEXT)";
    public static final String SQL_DELETE_USERS =
        "DROP TABLE IF EXISTS " + Users.TABLE_NAME;

    // Otras Sentencias ...
    // Las bases de datos, una para leer, una para escribir datos
    SQLiteDatabase escritor;
    SQLiteDatabase lector;
    public Ayudante(Context context){
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
}
```

Persistencia: SQLite

```
public boolean insertarUsuario(Usuario nuevo) {
    ContentValues values = new ContentValues();

    values.put(Users.COLUMN_NAME_NAME, nuevo.name);
    values.put(Users.COLUMN_NAME_DRINK, nuevo.drink);
    values.put(Users.COLUMN_NAME_SPORT, nuevo.sport);

    long inserted = escritor.insert(
            Users.TABLE_NAME,
            Users.COLUMN_NAME_NAME,
            values);

    if(inserted == -1) return false;
    return true;
}
```

Persistencia: SQLite

```
public List<Usuario> consultarUsuarios(){

    String[] columns = {Users._ID, Users.COLUMN_NAME_NAME,
        Users.COLUMN_NAME_DRINK, Users.COLUMN_NAME_SPORT};

    String selection = null; //Users.COLUMN_NAME_NAME + " like ?";
    String selectionArgs[] = null; //{"%a%"};
    String groupBy = null; //Users.COLUMN_NAME_SPORT;
    String having = null; //condición aritmética
    String orderBy = null; //Users._ID;
    String limit = null; //"10";

    Cursor results = lector.query(Users.TABLE_NAME,  columns,
        selection, selectionArgs, groupBy,  having,  orderBy, limit);
    // results ya es un cursor con los datos de regreso
}
```

Persistencia: Archivos

- Los directorios son de tipo File, así no sean archivos
- Opción correcta para imágenes, o elementos que puedan ser compartidos por red, incluso para borradores de correos, o borradores de configuraciones que puedan ser guardados para beneficio del usuario y evitar complicarlo con llenar de nuevo muchos datos.

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

Persistencia: Archivos Interna

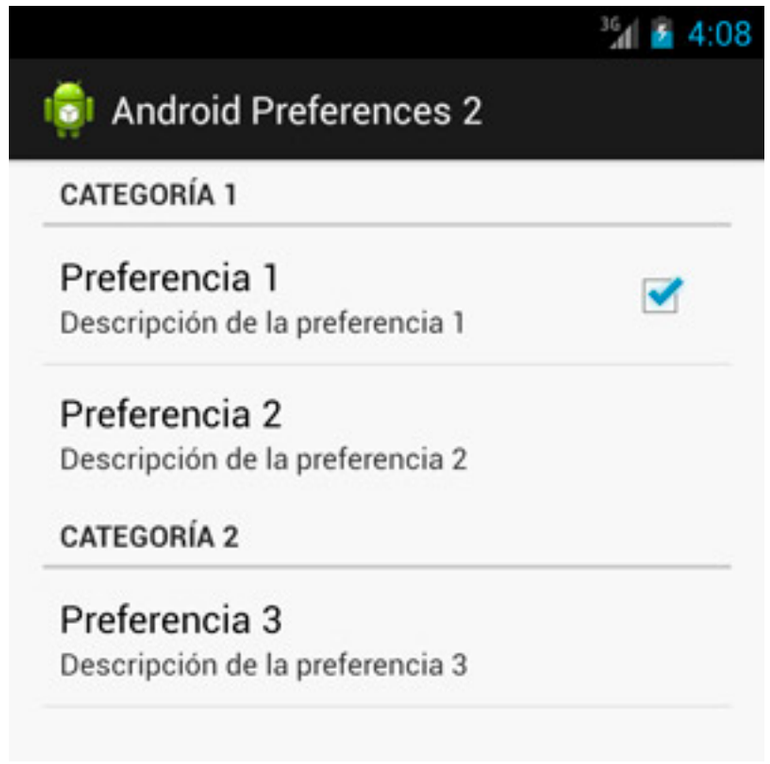
- Siempre está disponible, por defecto. Cambiar con android:installLocation
- Por defecto accede solo el app
- Cuando el usuario des-instala el app, se eliminan los archivos
- Es la mejor opción cuando queremos que ni el usuario u otras apps puedan acceder a los datos

Persistencia: Archivos Externa

- No siempre está disponible
- No tiene recursos, es WORLD_READABLE
- Solo se eliminan los archivos al des-instalar, cuando se guardan en getExternalDir()
- Es lo mejor cuando son archivos que queremos que se compartan con otras apps (¿fotos?)

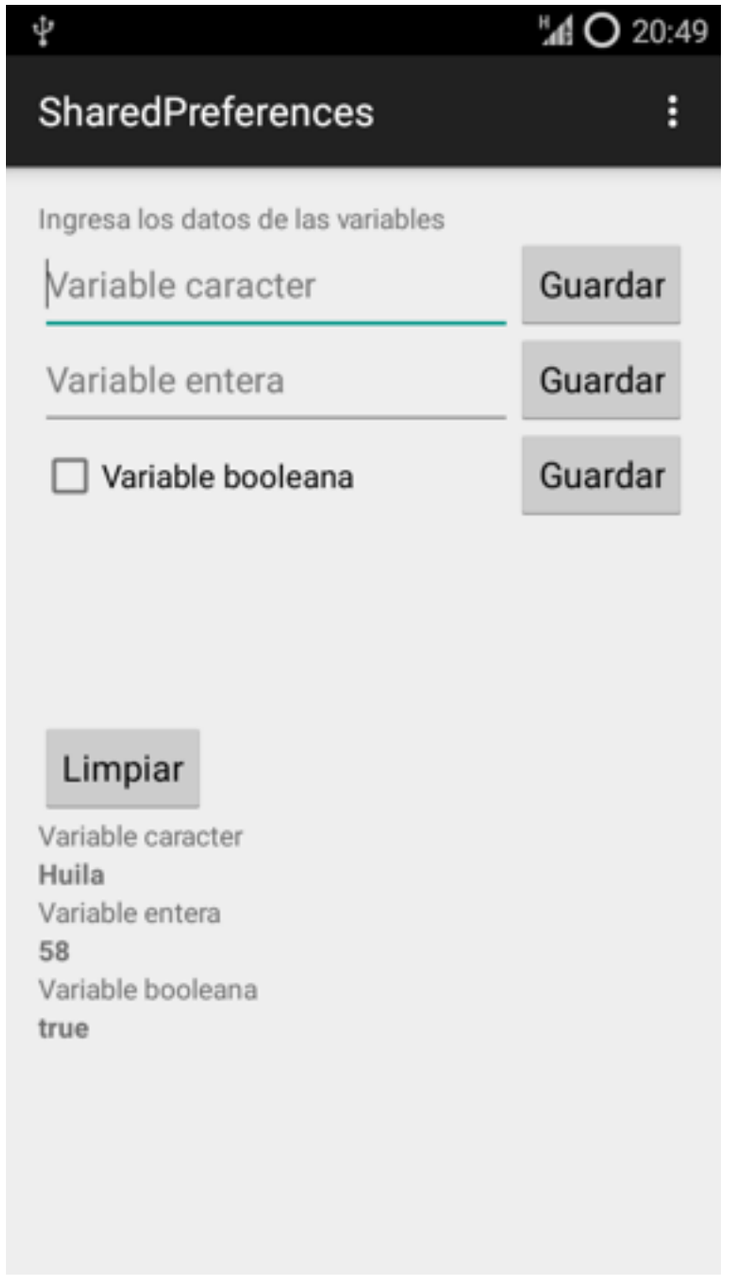
Persistencia: Preferencias compartidas

- Valores en mapa
- ruta-app/diccionario recuperado cada apertura
- getSharedPreferences() Vs getPreferences()
- MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, MODE_PRIVATE
- Editor: getEditor(), put…(), commit()



Persistencia: Preferencias compartidas

- Ejercicio tres tipos de variables
  - String
  - int
  - boolean



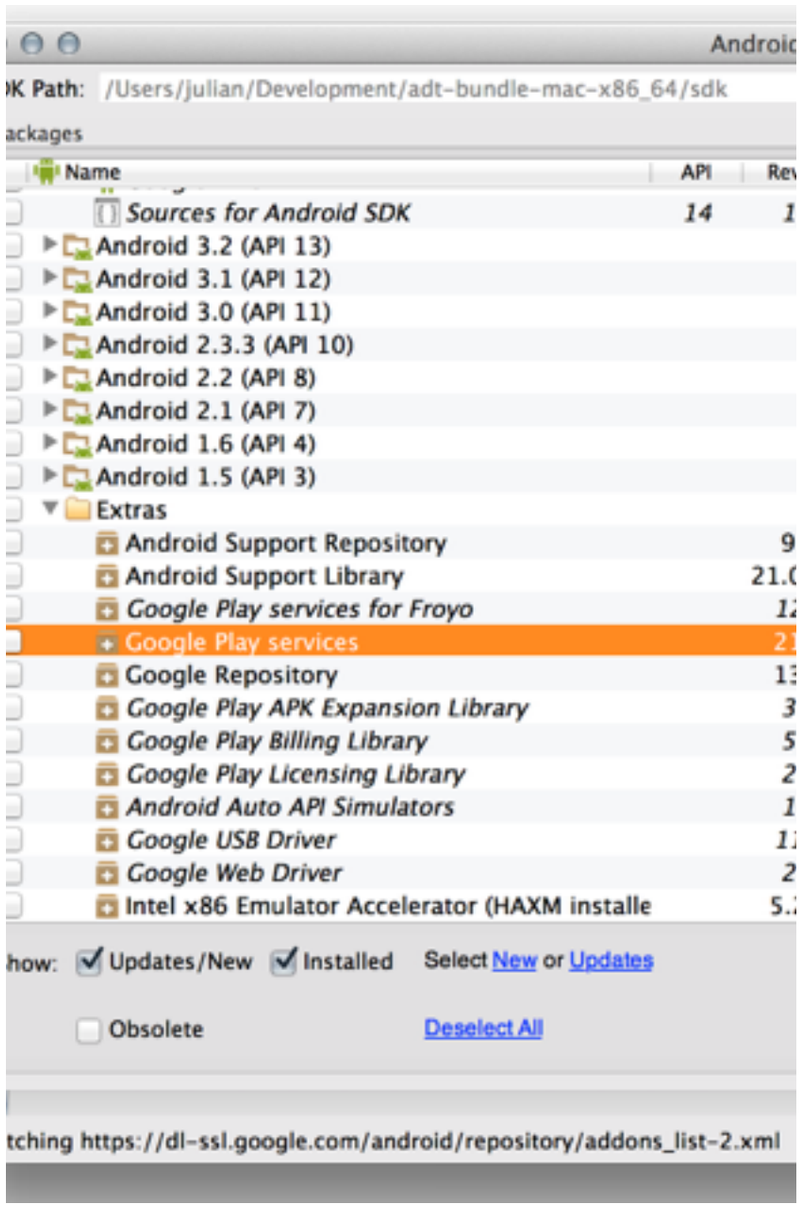
Geolocalización

- Fragmentos
- Conexión con la cuenta de desarrollado
- Firma SHA1

Geolocalización

- Manifiesto con permisos de acceso a sensores.

- Google Play Services



Geolocalización

```
<uses-permission
android:name="android.permission.INTERNET"/>
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission
android:name=
"com.google.android.providers.gsf.permission.READ_GSERVICES"/>

<!-- The following two permissions are not required to use
Google Maps Android API v2, but are recommended. -->
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Geolocalización: Java

```
import com.google.android.gms.maps.MapFragment;
/*…*/

MapFragment elMapaQueVamosAInyectar = new MapFragment();

FragmentManager fm = getActivity().getFragmentManager();
```

```
fm.beginTransaction().add(
   R.id.contenedor_mapa + mapaId,
   elMapaQueVamosAInyectar,
   “mapa")
.commit();
```

Geolocalización: XML

```
<fragment
android:id="@+id/map"
android:layout_width="match_parent"
android:layout_height="match_parent"
class="com.google.android.gms.maps.MapFragment" />

<FrameLayout android:id="@+id/contenedor_mapa"
android:layout_width="match_parent"
android:layout_height="match_parent"></FrameLayout>
```

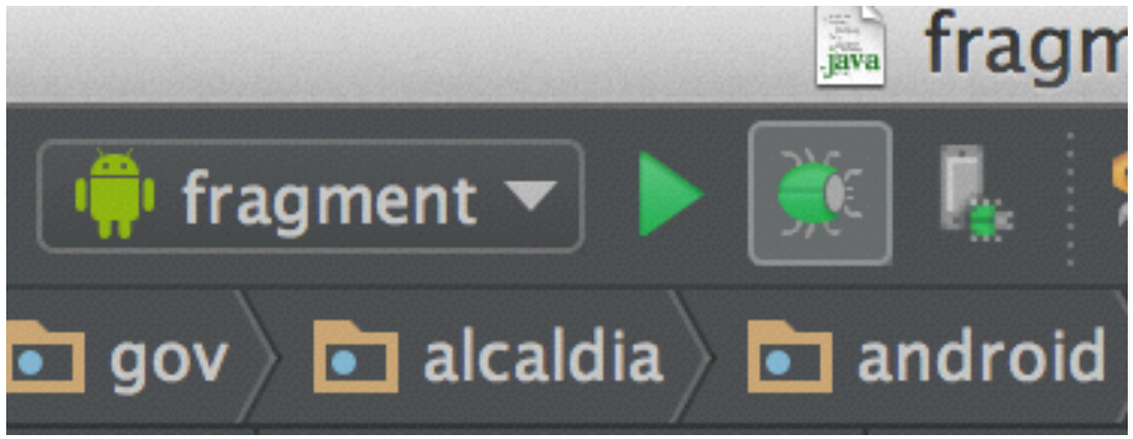
Agenda

TEMA	SESIÓN	TEMA	SESIÓN
Historia	1	Geolocalización	2
Workflow	1	IDE/Debug	3
“Hello World”	1	Fragmentos	3
Ciclo de vida	1	Interfaces	3
Action Bar	1	Notificaciones	4
XML Layouts	1	Consumo APIs	4
Menús	2	GCM	4
Persistencia	2	Q&A	1-4

Debug

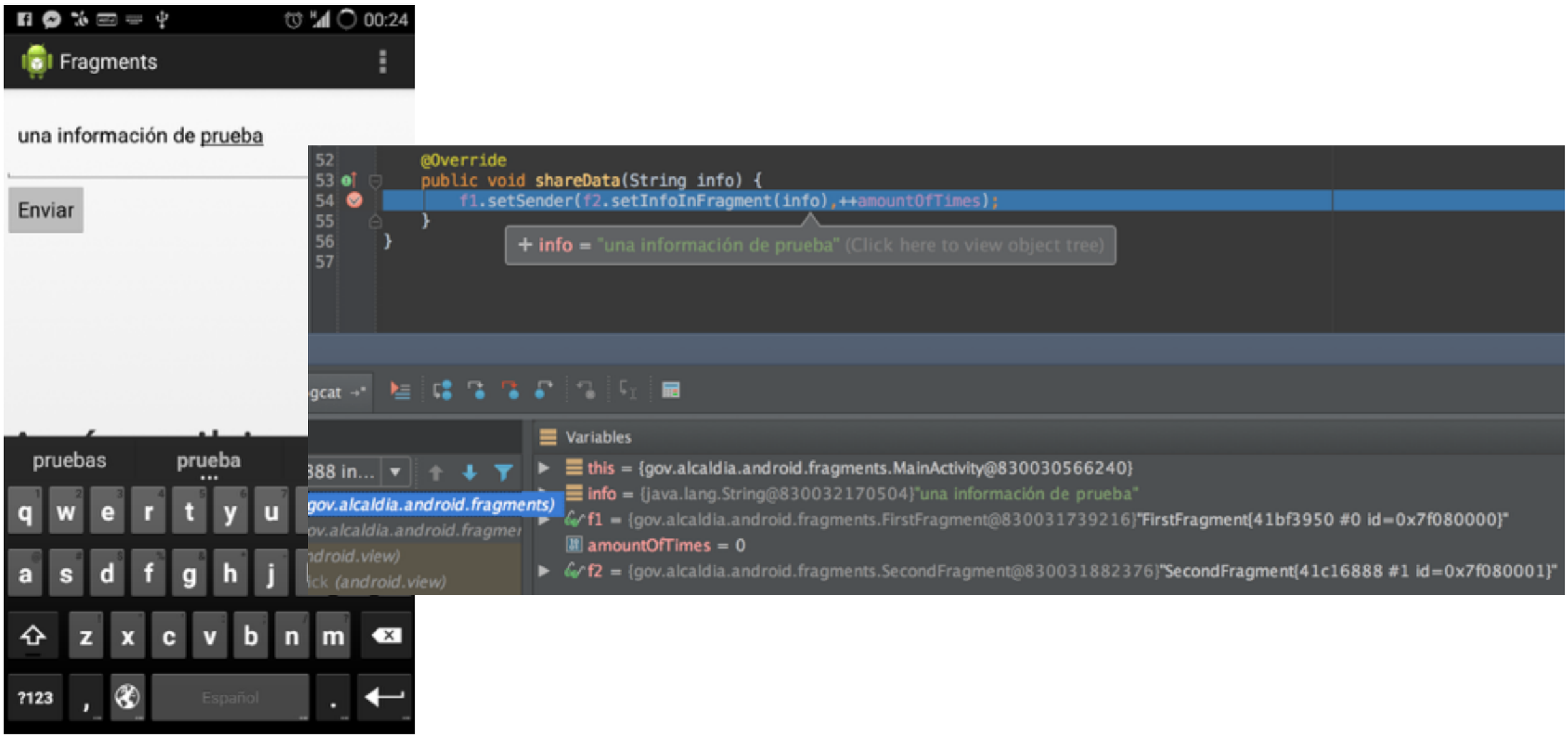
Escoger punto(s) de interrupción y ejecutar el APK con el insecto.

```
52          @Override
53          public void shareData(String info) {
54              f1.setSender(f2.setInfoInFragment(info),++amountOfTimes);
55          }
56      }
```



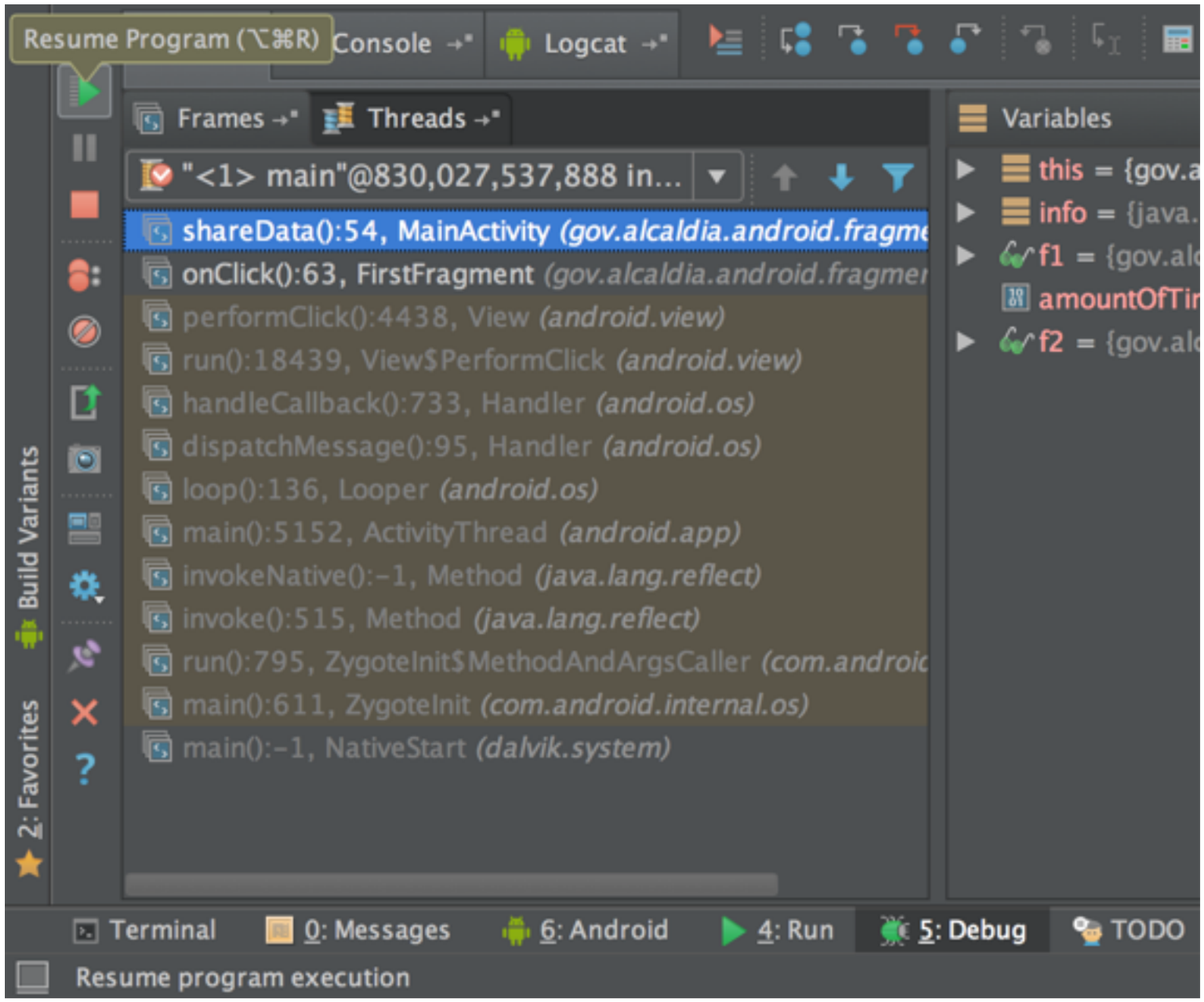
Debug

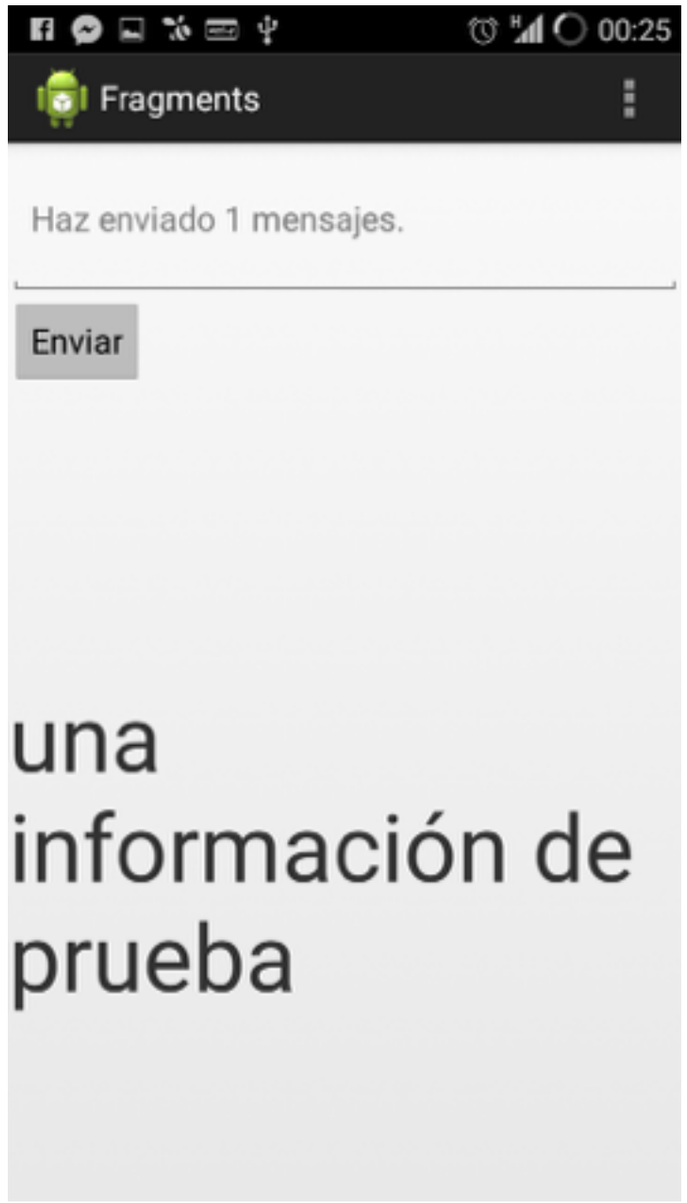
Llegar al punto de interrupción en el teléfono (ó emulador).



Debug

Inspeccionar de ser necesario, continuar.

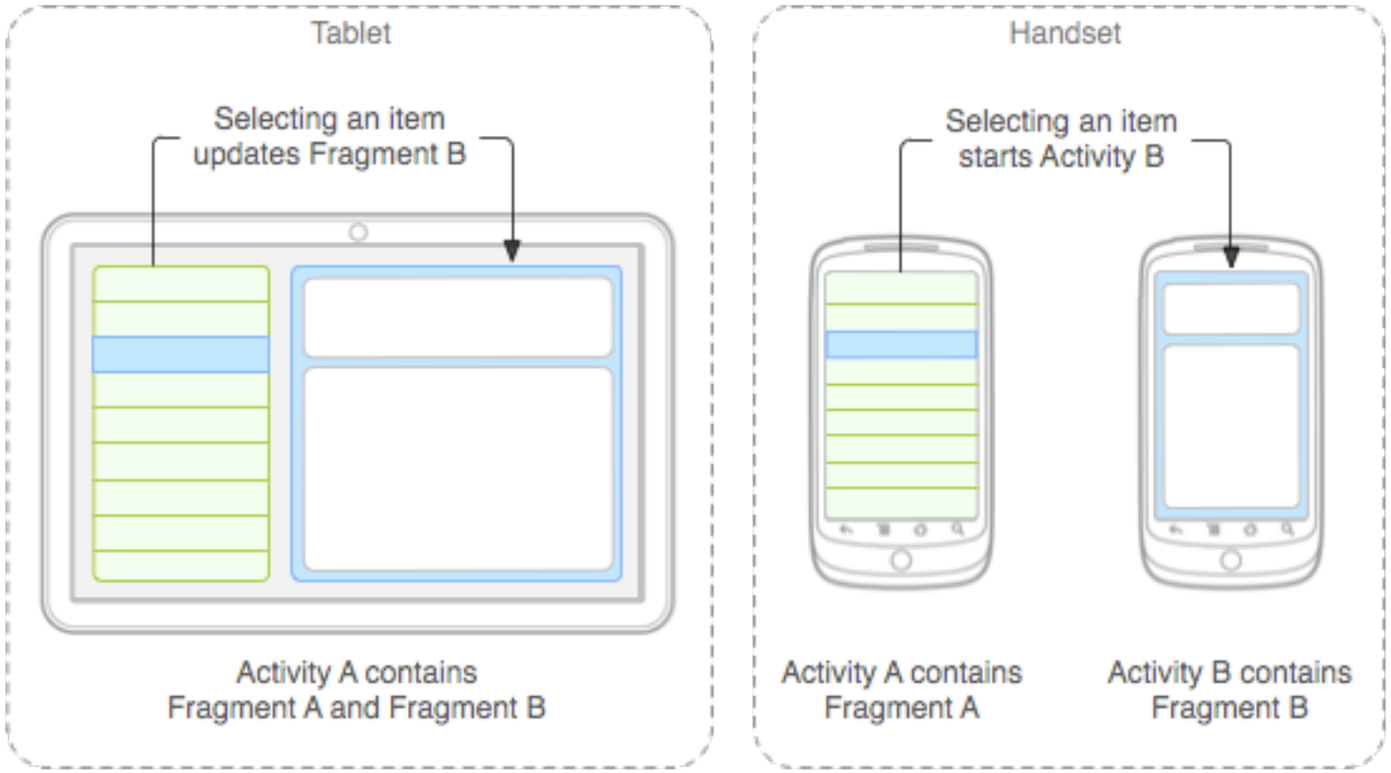


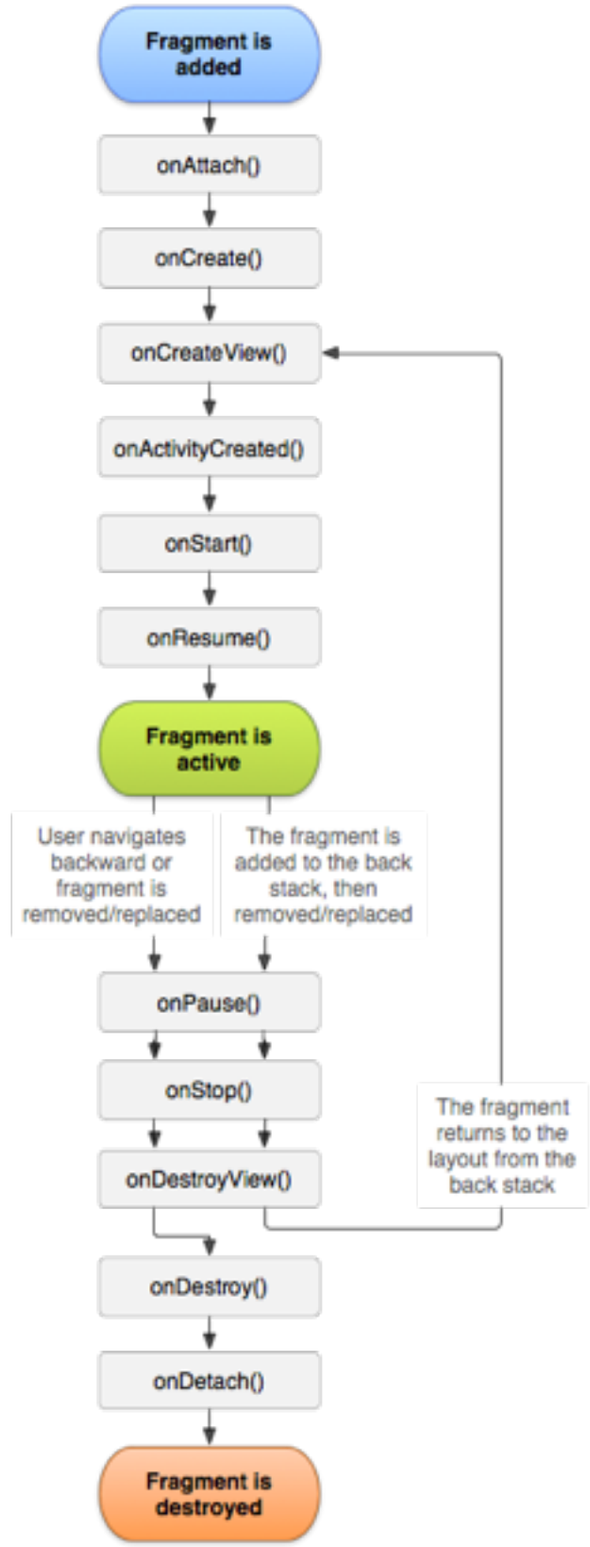


Fragmentos

- Nuevo estándar
- Re-utilización de código
- Ciclo de vida propio
- Transacciones
- Diseño multi-screen
  - Teléfonos
  - Tabletas
  - Wear

Fragmentos: Ciclo de vida





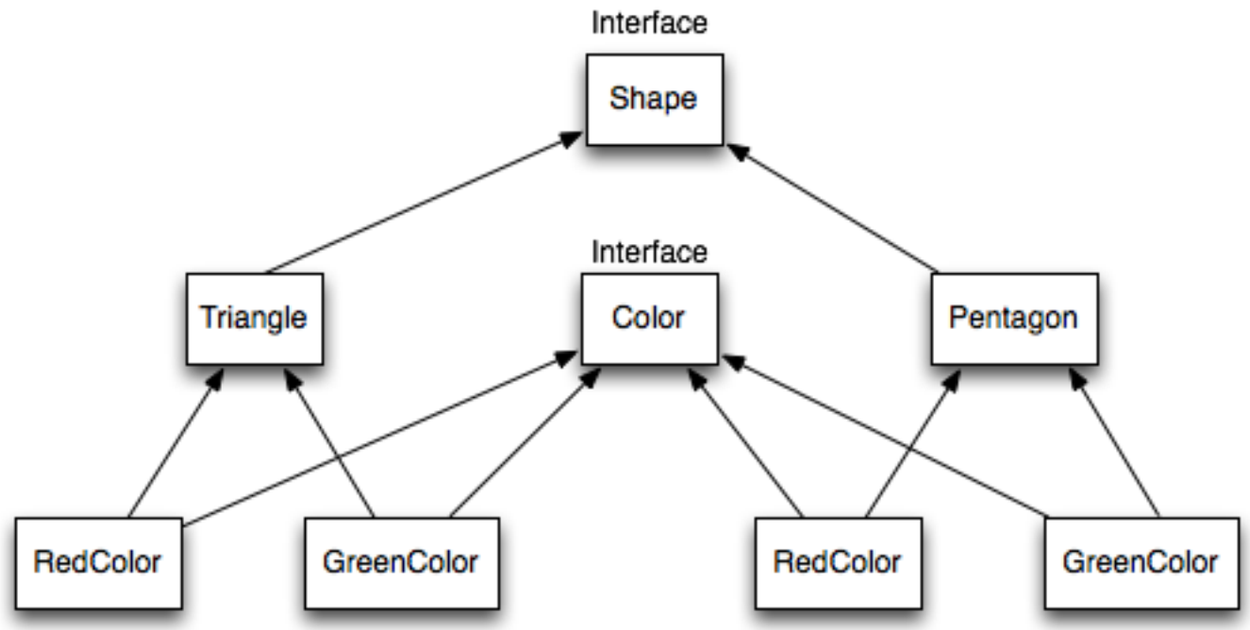
Fragmentos: Interfaces

- Manera segura de compartir información.

- getActivity() funciona desde un fragmento para acceder a otro, pero no de manera segura.

- La interfaz exige comunicación y favorece trabajo colaborativo.

- Estándar de métodos de comunicación.



Agenda

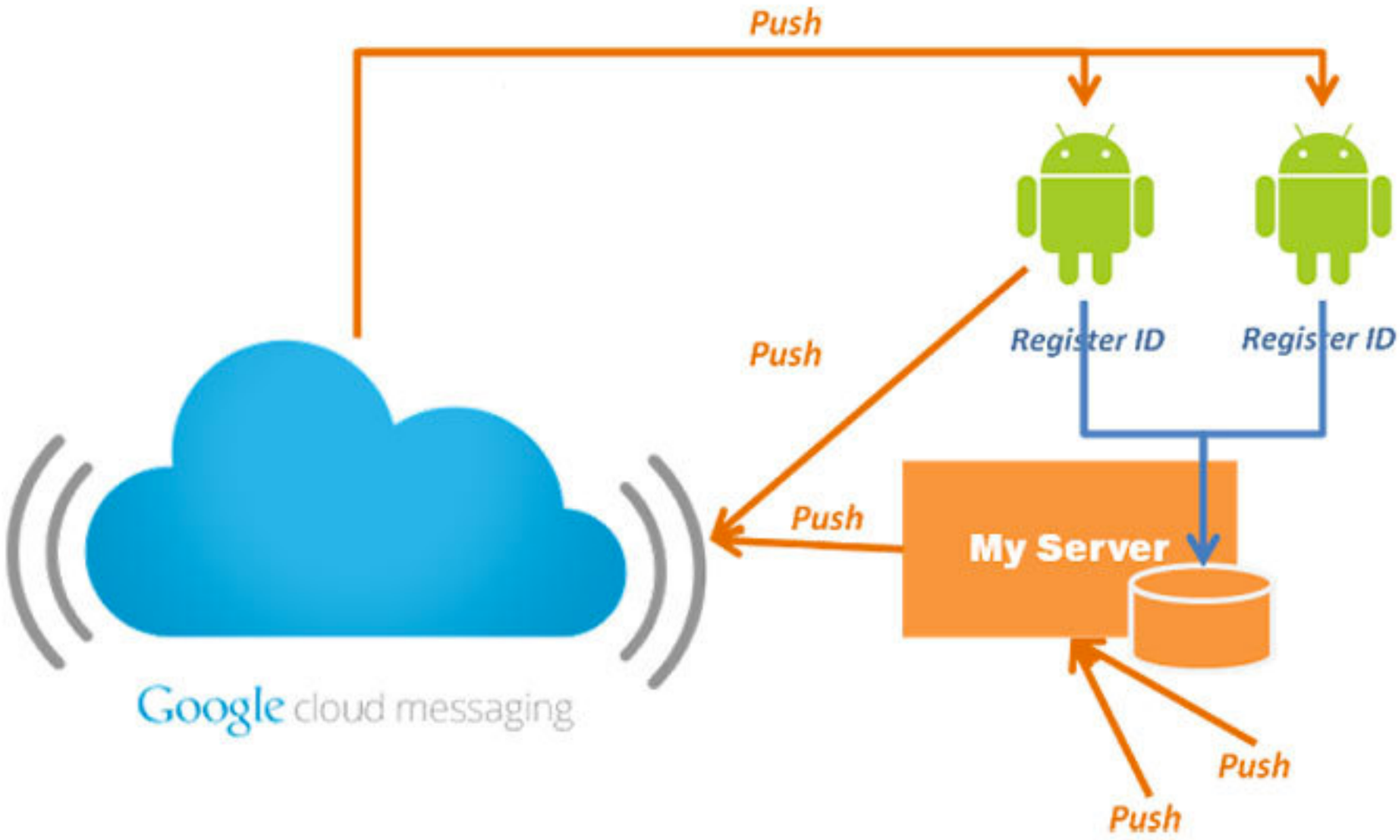
TEMA	SESIÓN	TEMA	SESIÓN
Historia	1	Geolocalización	2
Workflow	1	IDE/Debug	3
“Hello World”	1	Fragmentos	3
Ciclo de vida	1	Interfaces	3
Action Bar	1	Notificaciones	4
XML Layouts	1	Consumo APIs	4
Menús	2	GCM	4
Persistencia	2	Q&A	1-4

Notificaciones

Componentes Obligatorios:

- smallIcon
- title
- detailedText

GCM: Google Cloud Messaging



https://developer.android.com/google/gcm/index.html

GCM: Google Cloud Messaging

- SENDER_ID (API Console)
- Verificar SharedPreferences
- registerInBackground
  - Pedir un regId (registration ID) del server GCM
  - Alojar el ID en la DB propia
  - Alojar el ID en SharedPreferences

- BroadcastReceiver escuchando push
- IntentService que notifica