

# Introducción al Android

Sesion 1 Parte 2



# Introducción a Android





# ¿Qué es Android?

- Un Sistema Operativo basado en un kernel de Linux.
- Plataforma de desarrollo de aplicativos móviles.
- Conjunto de herramientas.
- Completo "entorno" de desarrollo de aplicativos.









Applications

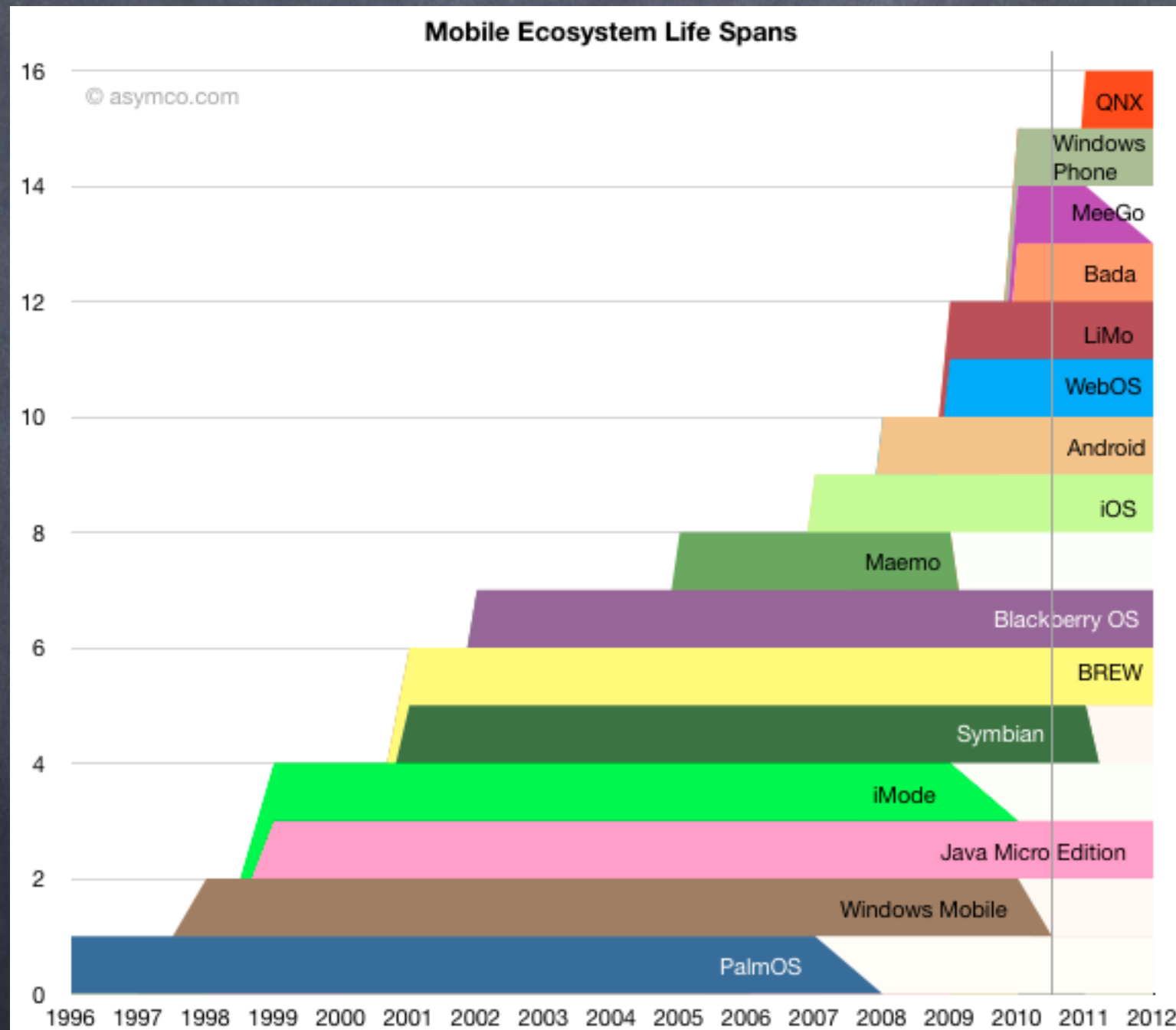
Application Framework

Libraries

Linux 2.6 Kernel



# Un poco de historia





# The Android Story

## Worldwide Android Market Share



Android, Inc. was founded in Palo Alto, California, United States by Andy Rubin, Rich Miner, Nick Sears and Chris White.

October, 2003



Google acquired Android Inc.

August 2005



The Open Handset Alliance, a consortium of several companies was formed.

5 November, 2007



Android Beta SDK Released.

12 November, 2007

0.50%







# ANDROID



**Cupcake**  
Android 1.5



**Donut**  
Android 1.6



**Eclair**  
Android 2.0/2.1



**Froyo**  
Android 2.2



**Gingerbread**  
Android 2.3



**Honeycomb**  
Android 3.0



**Ice Cream Sandwich**  
Android 4.0



**Jelly Bean**  
Android 4.1



**What Next?**





# Overview





# Arquitectura





# Kernel de Linux

- Plataforma madura
- Interfase con componentes hardware
- Es donde se encuentran los drivers
- Facilita la innovación





# Ambiente de ejecución

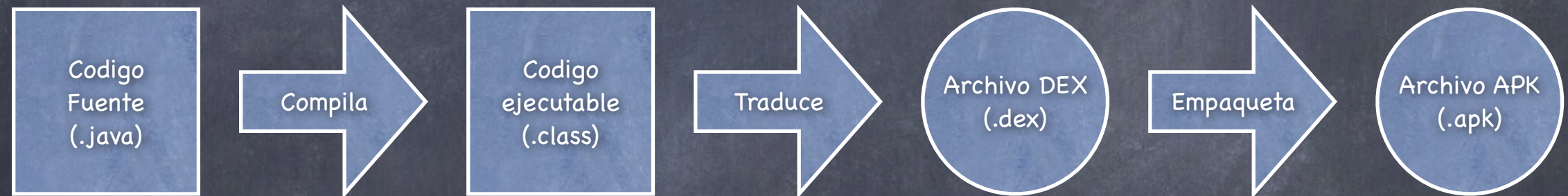




# Dalvik Virtual Machine

JVM optimizada para dispositivos móviles

Libre de problemas de licencias de Oracle



Cuidado con portar librerías entre JVM y DVM





# Librerías

- Surface Manager
  - Acceso a la pantalla
- OpenGL
  - Gráficos 2D y 3D
- SGL
  - Motor para gráficos 2D





# Librerías

- Media Framework
  - Reproducción de videos y de audio
- FreeType
  - Renderado de fonts
- SSL
  - Protocolo para transferencias seguras





# Librerías

- SQLite
  - BD ligera embebida en el dispositivo.
- Webkit
  - Motor para renderizado de páginas web
- libc (bionic)
  - Compilador de C





# Componentes Android





# Activity

- Administran la interacción con el usuario (UI)
- Se encargan de **controlar** las pantallas
  - carga (infla) con un layout
  - gestiona los eventos generados por el usuario
  - controla el ciclo de vida





# Activity

```
package com.devosinc.pruebaservicios;

+ import com.devosinc.pruebaservicios.services.PosicionService;

public class MainActivity extends Activity {
    Button butActivar;
    Button butDesactivar;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Installation.id(this);

        setContentView(R.layout.main);

        butActivar = (Button)this.findViewById(R.id.butActivar);
        butDesactivar = (Button)this.findViewById(R.id.butDesactivar);

        butActivar.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                startService(new Intent(MainActivity.this, PosicionService.class));
            }
        });

        butDesactivar.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
```





# Service

- No tiene UI
- Para ciclos de vida largos
- Responden a eventos normalmente no activados por el usuario



Deben tratar de ser tareas directas y concisas



# Service

```
import java.util.Timer;

public class PosicionService extends Service implements LocationListener{
    private Timer timer = new Timer();
    private static final long UPDATE_INTERVAL = 60000;

    LocationManager locationManager;
    private GeoPoint posicion;
    private float velocidad;

    public void onCreate() {
        super.onCreate();
        startService();
    }

    @Override
    public IBinder onBind(Intent arg0) {
        // TODO Auto-generated method stub
        return null;
    }

    private void startService(){
        Log.i(getClass().getSimpleName(), "Comenzamos el tracking.");

        locationManager = (LocationManager)this.getSystemService(Context.LOCATION_SERVICE);
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
            60000, 100, this);

        timer.scheduleAtFixedRate(new TimerTask() {
```





# Broadcast Receiver

- Respuesta a eventos globales (ejm recibo de llamada, llegada de SMS, etc).
- Se registran mediante el manifest o en tiempo de ejecución
- No tienen UI
- Utiliza intents



De corta duración. Si se necesita más tiempo, llamar un Service



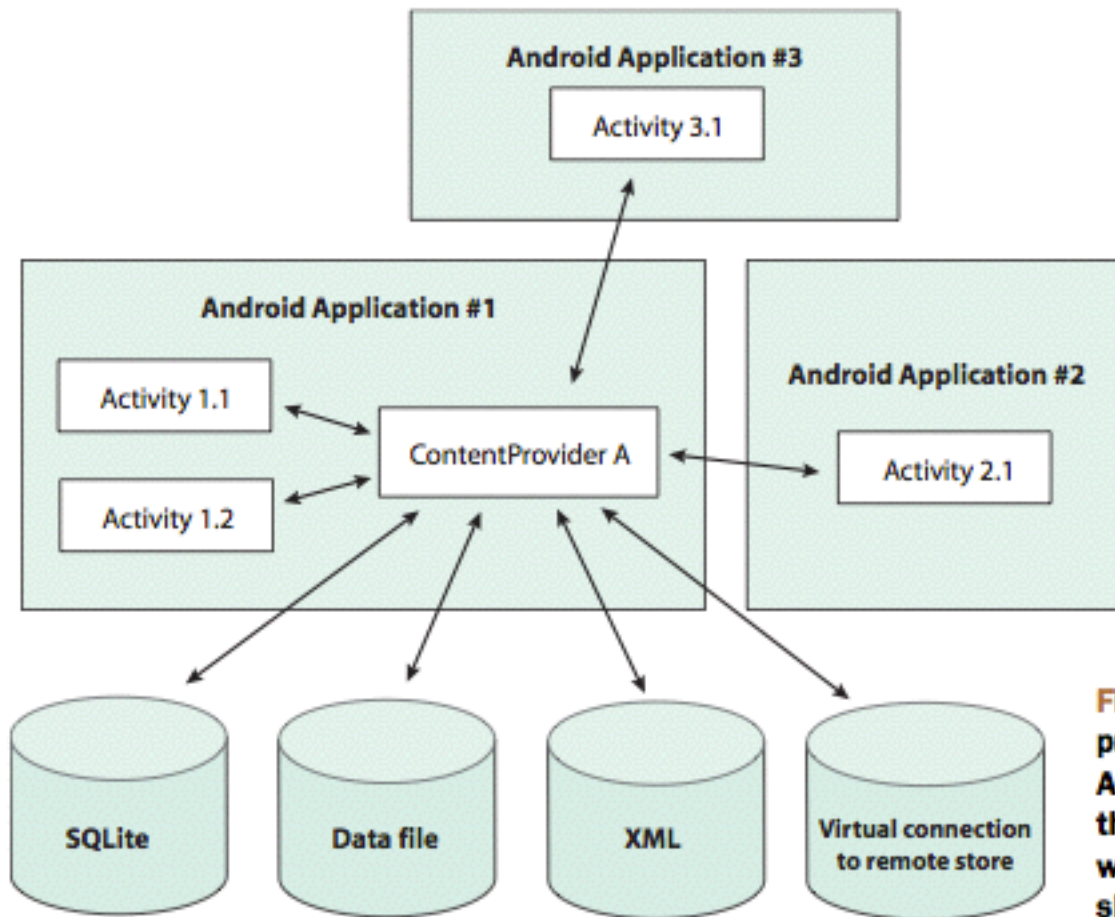
# Content Provider

- Compartir data entre aplicaciones.
- Los accesos pueden ser read, write o ambos.
- Acceso a través de URI.
- Manejo de tipos de datos tradicionales, así como binarios (imágenes).
- Pueden representar distintos orígenes de datos.





# Content Provider



**Figure 1.5** The content provider is the data tier for Android applications and is the prescribed manner in which data is accessed and shared on the device.





# Intents

- Declaración de una necesidad. Es el deseo de realizar una acción (activity) o un servicio (service).
- Mecanismo para invocar distintos componentes dentro del dispositivo (no solamente dentro de una app).
- Puede venir acompañado de data.

