Android Essentials

Introducción

Aplicaciones Móviles

FOLDER IT

¿Qué es una aplicación móvil?

Aquellas aplicaciones que son pensadas para ejecutar en dispositivos móviles como smartphones, smartwatches, tablets, etc.

Existen restricciones (Hardware limitado, Batería)

Y Extras (sensores que en una PC o servidor no se tienen)

Nativas vs Híbridas

FOLDER IT

Aplicaciones nativas

Son aplicaciones desarrolladas con el API y SDK de cada plataforma

Obtienen todas las ventajas de una plataforma, se instalan en el dispositivo, y pueden interactuar con el mismo y funcionar sin necesidad de que esté conectado a internet.

Son únicas para cada plataforma

Aplicaciones híbridas (Cordova, Phonegap, Ionic)

Son aplicaciones web, que se instalan en el teléfono.

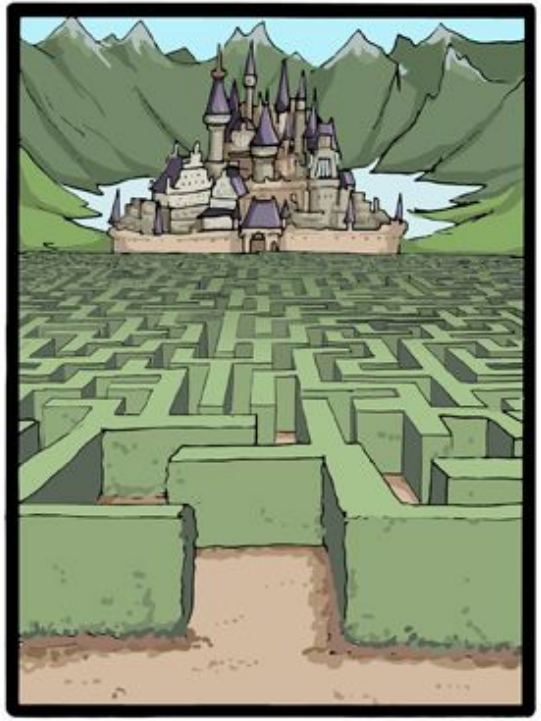
Son multiplataforma.

A pesar de ejecutar dentro de un navegador, pueden acceder a elementos del teléfono y además pueden trabajar en modo offline.

Nativas vs Híbridas

FOLDER IT

Develop a native app for each device and maintain several projects



Use a unique framework (Phonegap, Adobe Air, Appcelerator) and maintain only one project

Qué es Android

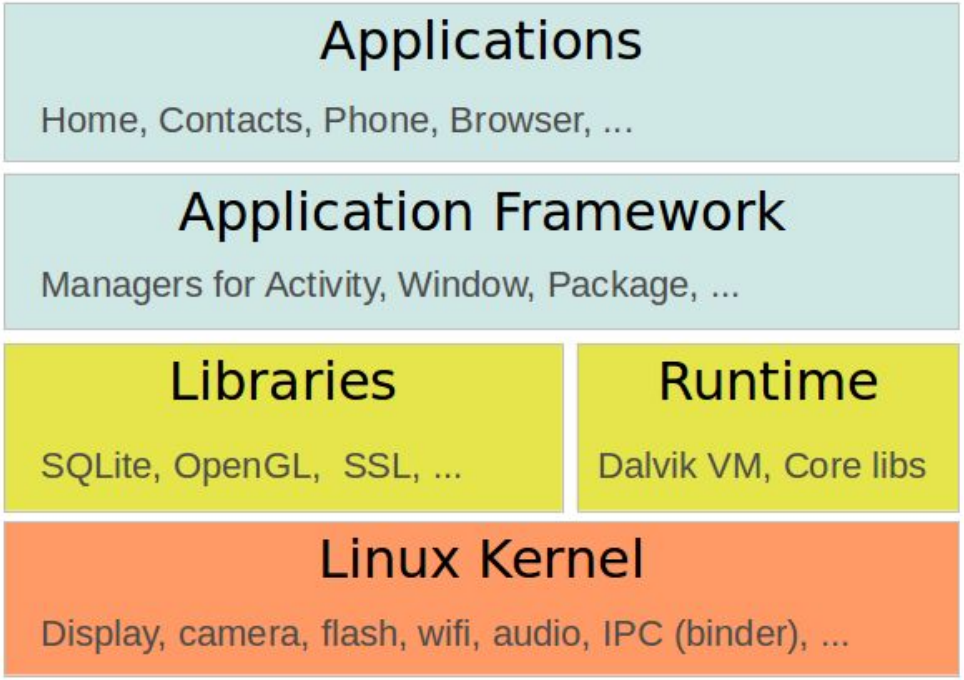
FOLDER IT

Android es un sistema operativo basado en el kernel de Linux.

El proyecto responsable de desarrollar Android es el AOSP (Proyecto Open Source Android) y es liderado por Google.

Lo podemos dividir en 4 capas.

Un desarrollador de Android comúnmente trabaja con las dos capas superiores



Que no es?

No es una implementación de Java ME.

No es la respuesta de Google al iPhone

Arquitectura

FOLDER IT

Aplicaciones - El Proyecto Abierto Android contiene varias aplicaciones por defecto como el navegador, camara, galería, reproductor de música, teléfono, etc

Application framework - Es una API que permite interacciones de alto nivel con el sistema desde las aplicaciones.

Librerías y runtime - Las librerías para varias funciones comunes del framework, como renderizado gráfico, almacenamiento de datos, navegación web. A su vez contiene el Runtime de Android (Dalvik, ART) como también las librerías de Java.

Kernel de Linux - Capa de comunicación con el hardware. (Drivers)

Ventajas

FOLDER IT

Es la plataforma más usada actualmente

Es open source (Apache 2.0 y GNU GPL)

Sus entornos de desarrollo son multiplataforma

Actualmente desde 2015 tenemos disponible el IDE Android Studio (Intellij IDEA)

Publicar una aplicación en el Play Store, es mucho más sencillo que en Apple Store

El lenguaje de programación es Java.

Desventajas

FOLDER IT

Android presenta algunos inconvenientes que es bueno tener en mente:

Fragmentación

la cantidad de dispositivos y versiones distinta de Android que puede tener un cliente pueden llegar a ser un problema

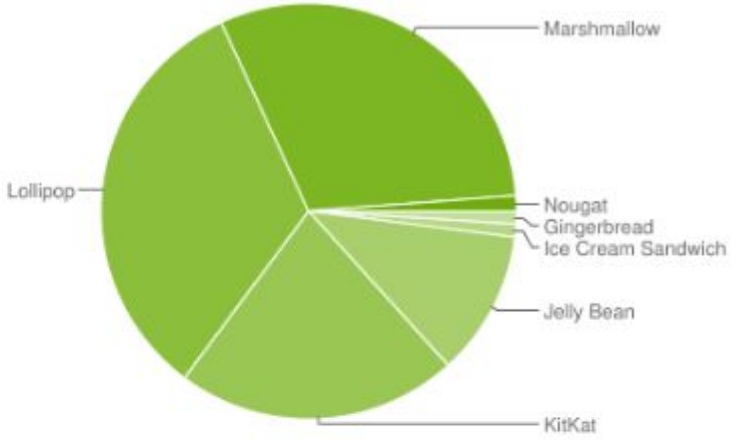
Fragmentación (2015)

FOLDER IT

Niveles de API

FOLDER IT

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.0%
4.1.x	Jelly Bean	16	4.0%
4.2.x		17	5.7%
4.3		18	1.6%
4.4	KitKat	19	21.9%
5.0	Lollipop	21	9.8%
5.1		22	23.1%
6.0	Marshmallow	23	30.7%
7.0	Nougat	24	0.9%
7.1		25	0.3%



Componentes principales

FOLDER IT

Application

Una App android tiene una clase `Application` que se instancia antes que cualquier componente y es el último componente en ser terminado

Si no se define explícitamente android lo crea

Activity

Es la representación visual de la App. Una App puede tener muchas Activities

Service

Estos ejecutan tareas sin tener una interfaz de usuario

Pueden comunicarse con otros componentes.

Componentes principales

FOLDER IT

Broadcast Receiver
Pueden ser registrados para escuchar mensajes del sistema e `Intents`
Son notificados por el sistema si un evento ocurre

Por Ejemplo: se puede recibir una notificación de que el sistema terminó de bootear, o que la bateria se esta cargando, etc

Content Provider
Definen una interfaz para que la App haga uso de datos
Puede ser usado para manejar datos internamente, como tambien para compartir información entre Apps.

Por Ejemplo: Android contiene una DB SQLite la cual se usa frecuentemente con un content provider. La DB va a almacenar los datos, los cuales se acceden mediante el content provider

Android Manifest

FOLDER IT

Manifest
Es un archivo que contiene metadatos inherentes a un conjunto de archivos.
Por Ejemplo, nombre, versión, etc

Android Manifest: `AndroidManifest.xml`

Los componentes, configuración y metadatos son descritos en este archivo
Todas las Actividades, Servicios y Content Providers deben ser declarados estáticamente aqui.
Los Broadcast Receiver pueden ser declarados estáticamente aquí o dinámicamente en tiempo de ejecución.

Este archivo es leído por el SO durante la instalación de la App. A partir de la evaluación del manifest se determinan las capacidades de la App
El sistema de build (Gradle) al compilar y empaquetar la App puede modificar el manifest.
Por Ejemplo, la versión de la App es provista por el script de Gradle.

Ejemplo de Manifest

FOLDER IT

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://schemas.android.com/tools"
    package="dam.isi.frsf.utn.edu.ar.delivery">
    <uses-feature android:name="android.hardware.camera" android:required="true" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application  android:allowBackup="true" android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name" android:supportsRtl="true"android:theme="@style/AppTheme"
        tools:ignore="AllowBackup" android:name="android.support.multidex.MultiDexApplication">
        <meta-data android:name="com.google.android.geo.API_KEY" android:value="@string/google_maps_key" />
        <activity android:name=".activity.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Archivos de Recursos

FOLDER IT

Android permite crear recursos estáticos como imágenes y archivos XML
Esto permite mantener separados los recursos del código fuente

Deben ser ubicados en el directorio `/res` en su subcarpeta correspondiente.
Cada subcarpeta depende del tipo de dato a almacenar

Drawables (`/res/drawables`)
Imagenes (png, jpg, etc), Imágenes vectoriales (svg), XML que escalan automáticamente con la densidad de píxeles del dispositivo

Valores Simples (`/res/values`)
Usado para definir strings, enteros, colores, dimensiones, estilos, Arrays estáticos
Por convención cada tipo se almacena en un archivo separado

```
res/values/strings.xml
res/values/colors.xml
res/values/dimens.xml
res/values/styles.xml
```

Archivos de Recursos

FOLDER IT

Layouts (`/res/layout`)
Son archivos XML que representan la vista (como html pero especifico para android)

Animaciones (`/res/animator`)
Animaciones definidas en XML que modifican la vista

Información sin formato (`/res/raw`)
Archivos varios guardados sin formato específico.
Se acceden a través de un objeto `InputStream`

Menús (`/res/menu`)
Define las acciones que se pueden realizar desde el menú en la barra de herramientas

El archivo R.java

FOLDER IT

A cada recurso importante en la carpeta `/res` se le da un ID
Android genera un archivo `R.java` el cual contiene valores de referencia autogenerados, estos son valores enteros estáticos.

Al agregar nuevos archivos de recursos,
la referencia correspondiente sera agregada al `R.java`
Este archivo no se debe tocar
El SDK provee métodos para acceder a los recuerdos correspondiente mediante su ID

Por ejemplo:
Para acceder a un String con ID `R.string.mString` en el código se usa el método definido en la clase Context: `getString(R.string.mString)`