



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE

Análisis de herramientas para mejorar el desarrollo de aplicaciones Android

CRISTOPHER NICOLÁS OYARZÚN ALTAMIRANO

Memoria de titulación para optar al Título de:
INGENIERO CIVIL INFORMÁTICO

Profesora Guía: Cecilia Reyes
Profesor Correferente: Chihau Chau

ABRIL DEL 2014.

Agradecimientos

Agradezco a Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit eu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

Ius id vidit volumus mandamus, vide veritus democritum te nec, ei eos debet libris consulatu. No mei ferri graeco dicunt, ad cum veri accommodare. Sed at malis omnesque delicata, usu et iusto zzril meliore. Dicunt maiorum eloquentiam cum eu, sit summo dolor essent te. Ne quodsi nusquam legendos has, ea dicit voluptua eloquentiam pro, ad sit quas qualisque. Eos vocibus deserunt quaestio ei.

Resumen

El crecimiento que ha tenido el sistema operativo Android es considerable. Existen más de un millón de aplicaciones disponibles en la tienda de Google y cada mes este número se ve incrementado. Es por ello que el proceso de desarrollo de aplicaciones ha ganado vital importancia. El objetivo de esta memoria es estudiar y comparar herramientas que ayuden a mejorar el desarrollo de aplicaciones Android de tal manera de proveer a los desarrolladores una guía práctica que les permita tomar mejores decisiones en el transcurso de un proyecto.

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetuer signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

Summary

Ius id vidit volumus mandamus, vide veritus democritum te nec, ei eos debet libris consulatu. No mei ferri graeco dicunt, ad cum veri accommodare. Sed at malis omnesque delicata, usu et iusto zzril meliore. Dicunt maiorum eloquentiam cum cu, sit summo dolor essent te. Ne quodsi nusquam legendos has, ea dicit voluptua eloquentiam pro, ad sit quas qualisque. Eos vocibus deserunt quaestio ei.

Índice general

1. Introducción	5
1.1. Definición del Problema	5
1.2. Objetivos	6
1.2.1. Objetivo principal	6
1.2.2. Objetivos específicos	6
1.3. Estructura del documento	6
2. Estado del Arte	8
2.1. Introducción a Android	8
2.1.1. Inicios de Android	8
2.1.2. Arquitectura	9
2.1.3. ¿Cómo las aplicaciones son compiladas?	11
2.1.4. Tipos de dispositivos	12
2.1.5. Versiones	13
2.2. Problemas al desarrollar en Android	14
2.2.1. Fragmentación a nivel de software	14
2.2.2. Fragmentación a nivel de hardware	16
2.2.3. Otros problemas	18
3. Herramientas Actuales	19
3.1. Herramientas de Testing	19
3.1.1. JUnit	19
3.1.2. EasyMock	19
3.1.3. PowerMock	20
3.1.4. Mockito	20
3.1.5. Espresso	20
3.1.6. Robotium	20
3.1.7. Robolectric	20
3.1.8. Spoon	21
3.2. Herramientas de Crashes	21
3.2.1. Crittercism	21
3.2.2. Bugsense	21
3.2.3. Crashlytics	21
3.2.4. ACRA	22

3.2.5. Google Analytics	22
4. Análisis Comparativo	23
5. Implementación en entorno real	24
6. Conclusión	25

Capítulo 1

Introducción

1.1. Definición del Problema

Android es una sistema operativo emergente de código abierto, diseñado especialmente para dispositivos móviles, el cual fue presentado el año 2007. El crecimiento que ha tenido los últimos años ha sido considerable, dominando el mercado ampliamente, existiendo más de mil millones de dispositivos activados en todo el mundo.

El gran problema que ha tenido que enfrentar la gente que desarrolla aplicaciones para Android es la fragmentación generada por la inmensa cantidad y variedad de dispositivos existentes, entre los que se encuentran smartphones, notebooks, netbooks, tablets, televisores, etc. Además, existen variadas versiones del mismo sistema operativo en uso. Esto conlleva muchos problemas a la hora de diseñar y desarrollar aplicaciones ya que es casi imposible poder probar nuestra aplicación en cada uno de los dispositivos para los cuales estará disponible. Lo más probable es que existirán problemas de interfaz gráfica si no estamos listos para soportar variadas resoluciones y tamaños de pantalla. También es necesario tener un buen sistema de reporte de crashes ya que muchas veces, por más que nuestro código funcione de forma correcta en un dispositivo con Android 4.3, en el mismo dispositivo con Android 4.0.4 se puede comportar de forma distinta. Además, cada dispositivo tiene diferentes capacidades con respecto a: procesador, número de núcleos, memoria, RAM, batería, entre otras cosas. Si bien, estos son sólo algunos de los problemas que se deben enfrentar, existen otros más, tales como: la distribución de versiones alphas y betas, tracking de eventos, mejoras de performance, testing, etc.

Hoy en día existen variadas herramientas que ayudan a afrontar muchos de estos problemas. El rápido crecimiento de Android ha estado acompañado de la aparición de una gran cantidad de librerías y proyectos opensource con diversas soluciones. Estas herramientas se dan a conocer a través de las redes sociales

o comunidades de programadores. Debido a que estas librerías se encuentran dispersas es difícil saber cual de todas elegir, ya que se desconocen los pros y contras de cada una de ellas. Esto provoca que muchas veces, por desconocimiento o falta de tiempo, no se realice una decisión informada y se use la primera librería encontrada.

1.2. Objetivos

A continuación se presentan la lista de objetivos que se desean abarcar en este trabajo:

1.2.1. Objetivo principal

- Estudiar y comparar herramientas que ayudan a mejorar el desarrollo de aplicaciones Android de tal manera de proveer a los desarrolladores una guía práctica que les permita tomar mejores decisiones en el transcurso de un proyecto.

1.2.2. Objetivos específicos

- Identificar los distintos problemas existentes durante el desarrollo de aplicaciones Android.
- Estudiar y clasificar las herramientas actuales en base a los distintos problemas que buscan solucionar.
- Realizar un análisis comparativo para cada una de las categorías elegidas.
- Implementar las mejores herramientas en un entorno real de desarrollo.

1.3. Estructura del documento

El capítulo 2 corresponde al Estado del Arte. En él se realizará una introducción a Android, explicando a grandes rasgos sus inicios, arquitectura, tipos de dispositivos, entre otras cosas. Además estudiarán los problemas más comunes que se presentan en las distintas fases del desarrollo de aplicaciones Android.

En el capítulo 3 se presentará un listado clasificado con las distintas herramientas para mejorar el desarrollo de aplicaciones. Se examinará cada una de estas, lo que permitirá tener un panorama general de las fortalezas y debilidades que poseen.

En el capítulo 4 se realizará un análisis y se compararan algunas herramientas para poder concluir que se debe usar y para qué casos.

En el capítulo 5 se llevará a cabo una implementación de las mejores herramientas en un entorno real de desarrollo.

En el capítulo 6 posee las conclusiones obtenidas apartir de los análisis y las implementaciones realizadas

Capítulo 2

Estado del Arte

En este capítulo se dará a conocer una breve descripción del sistema operativo Android. Se comenzará con una introducción y se hablará de sus inicios, su arquitectura y la evolución que ha tenido con el tiempo. Además se hablará sobre los problemas más comunes al momento de comenzar a desarrollar una aplicación para Android.

2.1. Introducción a Android

Android es un sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles táctiles, tales como smartphones y tablets. A continuación se darán a conocer algunos detalles sobre sus inicios, arquitectura y evolución con el tiempo.

2.1.1. Inicios de Android

Android, Inc. fue fundada en Palo Alto, California en Octubre del 2003 por Andy Rubin, Rich Miner, Nick Sears and Chris White. Su objetivo era desarrollar dispositivos móviles más inteligentes, que estuvieran más enfocados en la localización del dueño y en distintas preferencias.

Google compró a Android Inc. el 17 de Agosto del 2005. Poco se sabía sobre esta compañía para ese entonces ya que estuvo funcionando de forma secreta, sin dar a conocer muchos detalles sobre lo que desarrollaban. Muchos asumían que Google estaba planeando entrar al mercado de dispositivos móviles. De ahí en adelante los esfuerzos de Google se enfocaron en conversaciones con fabricantes y carriers, con la promesa de proveer de un sistema flexible y actualizable.

Sin embargo, la aparición del iPhone el 9 de Enero del 2007 tuvo un efecto disruptivo en el desarrollo de Android. Hasta el momento se contaba con un prototipo, el cuál se acercaba más a lo que podría ser un teléfono BlackBerry,

sin pantalla táctil y con un teclado físico. Por lo que se comenzó inmediatamente un trabajo de reingeniería del sistema operativa y del prototipo para que fuese capaz de competir con el iPhone.

El 6 de Noviembre del 2007 fue fundada la Open Handset Alliance, una alianza comercial liderada por Google con compañías tecnológicas como HTC, Sony y Samsung, operadores de carriers como Nextel y T-Mobile y fabricantes de chips, con el objetivo de desarrollar estándares abiertos para dispositivos móviles. El primer smartphone disponible que funcionaba sobre Android fue el HTC Dream, lanzado el 22 de Octubre del 2008.

2.1.2. Arquitectura

La arquitectura del sistema Android, también llamado stack, se puede apreciar en la figura 2.1 y está compuesta por cuatro capas:

- **Kernel de Linux:** La capa más profunda es su núcleo en Linux, un sistema operativo abierto, el cuál es portable y seguro. Para cada pieza de hardware, como la cámara o el bluetooth, existe un driver dentro del kernel, que permitirá a la capa superior hacer uso de ella, por lo que funciona como una capa de abstracción. El kernel además se encarga de la gestión de los diversos recursos del dispositivo, como la energía o la memoria, elementos de comunicación, procesos, etc.
- **Bibliotecas:** La segunda capa en el stack contiene bibliotecas nativas, las cuáles están escritas en C o C++, y son compiladas para la arquitectura específica del dispositivo. En la mayoría de los casos el fabricante es quien se encarga de instalarla en su dispositivo. Las bibliotecas incluidas en esta capa son: el motor gráfico OpenGL, el sistema de gestión de base de datos SQLite, cifrado de comunicaciones SSL, motor de manejo de tipos de letra FreeType, entre otras.

El entorno de ejecución de Android también está compuesto por bibliotecas, por lo que no se considera una capa. Debido a las limitaciones de los dispositivos en los que debe funcionar, Google decidió crear la máquina virtual Dalvik, que funciona de forma similar a la máquina virtual de Java. Esta permite crear aplicaciones con un mejor rendimiento y menor consumo de energía, lo que es muy importante en dispositivos móviles. Además en el entorno de ejecución se incluyen la mayoría de las bibliotecas básicas de Java.

- **Marco o Framework de Aplicaciones:** La tercera capa está compuesta por todas las clases y servicios que se utilizan al momento de programar aplicaciones. Los componentes que posee son:
 - **Administrado de actividades (Activity Manager):** Gestiona la pila de actividades de la aplicación, como también su ciclo de vida.

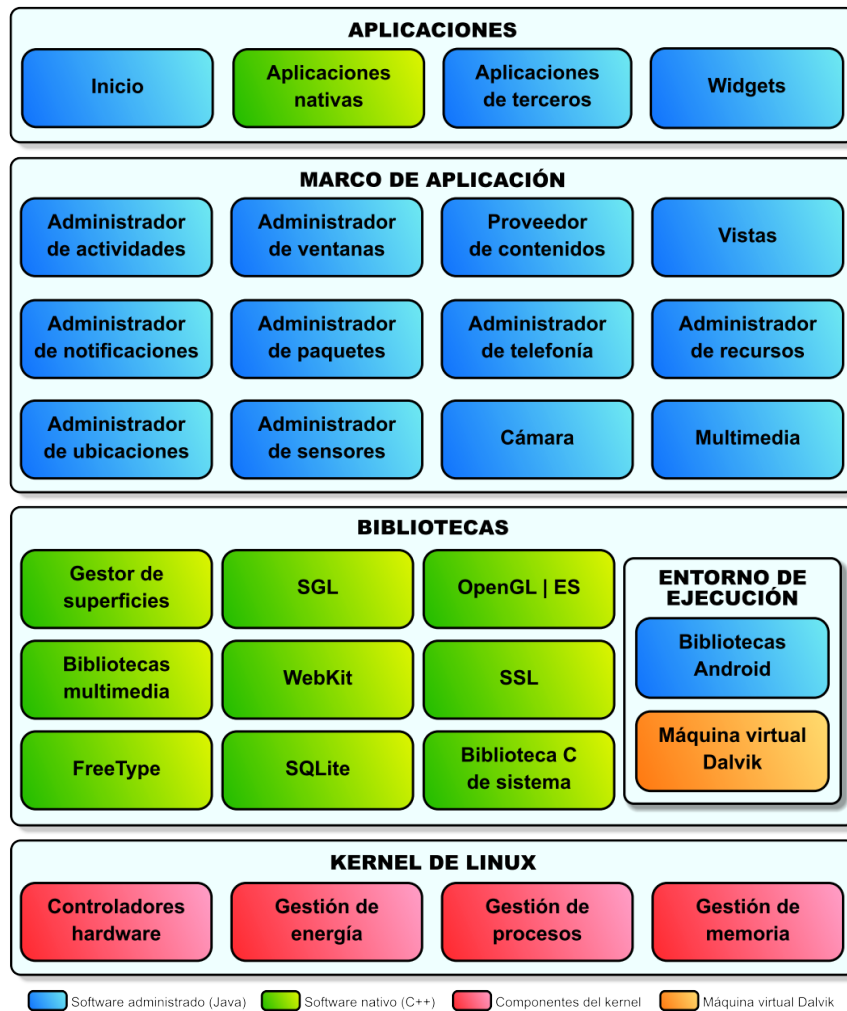


Figura 2.1: Arquitectura de Android, compuesta por cuatro capas

- **Administrador de ventanas (Windows Manager):** Organiza lo que se mostrará en pantalla. Crea las superficies en la pantalla, que posteriormente estarán ocupadas por las actividades.
- **Proveedor de contenidos (Content Provider):** Encapsula los datos que pueden ser compartidos por las aplicaciones, facilitando la comunicación entre estas.
- **Vistas (Views):** Son los elementos que nos permitirán construir las interfaces de usuario, como listas, botones, textos, hasta otros elementos más avanzados como visores de mapas.

- **Administrador de notificaciones (Notification Manager):** Provee los servicios que notifican al usuario, mostrando alertas en la barra de estado. También permite activar el vibrado, reproducir alertas de sonido y utilizar las luces del dispositivo.
 - **Administrador de paquetes (Package Manager):** Gestiona la instalación de nuevos paquetes y además permite obtener información sobre los que ya están instalados.
 - **Administrador de telefonía (Telephony Manager):** Permite realizar llamadas, como también el envío y recepción de SMS.
 - **Administrado de recursos (Resource Manager):** A través de este administrador se podrá acceder a los elementos que no forman parte del código, como imágenes, sonidos, layouts, etc.
 - **Administrado de ubicaciones (Location Manager):** Permite obtener la posición geográfica actual del dispositivo a través de GPS o redes.
 - **Administrado de sensores (Sensor Manager):** Permite la manipulación de distintos sensores del dispositivo, como el acelerómetro, giroscopio, brújula, sensor de proximidad, etc.
 - **Cámara:** Permite el uso de la cámara del dispositivo para la obtención de fotografías o vídeos.
 - **Multimedia:** Permite la visualización y reproducción de imágenes, vídeos y audio.
- **Aplicaciones:** En esta capa se encuentran todas las aplicaciones del dispositivo, tanto las preinstaladas, como aquellas instaladas por el usuario. También está la aplicación principal del sistema, el Inicio o launcher, desde donde se inician todas las aplicaciones.

2.1.3. ¿Cómo las aplicaciones son compiladas?

Al comenzar a desarrollar una aplicación de Android, generalmente se crea un proyecto usando un IDE (*Integrated Development Environment*) como Eclipse o Android Studio. El proyecto contendrá código fuente en Java y recursos. Cuando se compila el proyecto lo que ocurre es que se generan los Bytecode Java (archivos .class) en base a nuestro código fuente Java (archivos .java). Luego se compilan estos archivos .class a archivos ejecutables Dalvik (archivos .dex), los cuales pueden ser ejecutados por la máquina virtual Dalvik que está disponible en todos los dispositivos Android.

Al compilar un proyecto se colocan los archivos .dex y el resto de los archivos del proyecto en un archivo llamado APK (*Android Package*). Este contiene todos los archivos necesarios para ejecutar la aplicación, incluyendo los archivos .dex, recursos compilados, recursos sin compilar, y una versión binaria del

Android manifest.

El *Android manifest* es un archivo que especifica información esencial que el sistema debe tener antes de ejecutar la aplicación. Toda aplicación debe tener este archivo de forma no binaria en su proyecto.

Por razones de seguridad todas las aplicaciones de Android deben ser firmadas digitalmente con un certificado.

Finalmente el *Android debug bridge (ADB)* permiten que el IDE se comunique con un dispositivo físico de Android o un emulador.

2.1.4. Tipos de dispositivos

En el sitio web de Android [2], se pueden apreciar los dos tipos de dispositivos más populares de la plataforma, los smartphones y las tablets (Figura 2.2). Sin embargo, debido a que el código de Android es de código abierto, este puede ser personalizado para que funcione con otros tipos de dispositivos electrónicos.



Figura 2.2: Últimos smartphones y tablets destacadas en el sitio de Android.

A continuación se listan los otros dispositivos que cuentan con Android:

- Lectores de libros.
- Cámaras.
- Sistemas en vehículos.
- Casas inteligentes.
- Consolas de videojuegos.
- Televisores inteligentes.
- Relojes inteligentes.

2.1.5. Versiones

En la figura 2.3 se detallan las distintas versiones que ha tenido Android. La primera versión comercial fue lanzada en Septiembre del 2008. Android está bajo constante desarrollo por parte de Google y de la Open Handset Alliance, contando con un gran número de actualizaciones desde su lanzamiento.

Desde Abril del 2009, los nombres de las versiones de Android han estado relacionados con postres y dulces, y además han seguido un orden alfabético. El orden es Cupcake (1.5), Donut (1.6), Eclair (2.0-2.1), Froyo (2.2-2.2.3), Gingerbread (2.3-2.3.7), Honeycomb (3.0-3.2.5), Ice Cream Sandwich (4.0-4.0.4), Jelly Bean (4.1-4.3), y KitKat(4.4). El 3 de Septiembre del 2013, Google anunció que existían un billón de dispositivos activos usando el sistema operativo Android en todo el mundo. La actualización más reciente de android fue KitKat 4.4, el cual fue lanzado para dispositivos comerciales el 22 de Noviembre del 2013.



Figura 2.3: Versiones de Android.

Al comenzar el desarrollo de una aplicación Android, se debe decidir cual va a ser la API mínima a la cuál se dará soporte. Esto tendrá repercusiones al momento de que un usuario desee instalar la aplicación, ya que si su dispositivo cuenta con una versión como Froyo o Eclair, lo más probable es que no pueda instalar prácticamente ninguna de las aplicaciones disponibles en Google Play, la tienda en que se encuentran todas las aplicaciones que suben los desarrolladores.

Android actualiza mes a mes las estadísticas relativas al número de dispositivos que tienen cada versión del sistema operativo [1]. Esto ayuda a tener una guía sobre cuál va a ser la API mínima soportada. En la figura 2.4 se muestran las estadísticas correspondientes al mes de Abril. Esta información es recolectada durante los últimos 7 días de cada mes, terminando el 1 de Abril del 2014. Además se ignoran las versiones que tienen menos de un 0.1 %. Se puede apre-

ciar que el sistema operativo que hoy en día es dominante corresponde a Jelly Bean con más de un 60 %. El nuevo sistema operativo KitKat tiene sólo un 5.3% debido principalmente a que los operadores y fabricantes aún no tienen listas sus versiones personalizadas de KitKat, en las que pueden incluir nuevas funcionalidades o quitar lo que estimen conveniente.

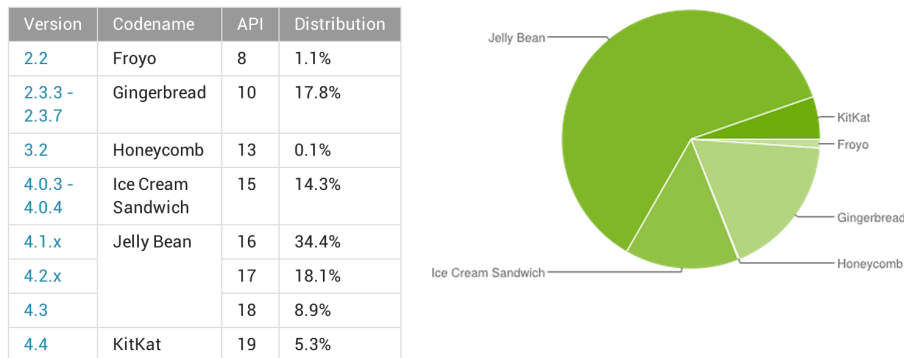


Figura 2.4: Estadísticas relativas al número de dispositivos que tiene cada versión de Android en Abril del 2014.

2.2. Problemas al desarrollar en Android

Ahora que ya se han dado a conocer aspectos básicos sobre Android, se puede profundizar en los problemas más comunes que se enfrentan al desarrollar aplicaciones.

2.2.1. Fragmentación a nivel de software

Como ya se mencionó anteriormente, existen muchos sistemas operativos de Android vigentes hoy en día. Esto priva al desarrollador de muchas funciones útiles al momento de programar su aplicación, ya que se debe establecer una API mínima. En base a las estadísticas que provee Android, la mayoría de los desarrolladores decide dar soporte desde Gingerbread en adelante. Si el desarrollador desea utilizar métodos de una API superior a la de Gingerbread, debe especificar en el código fuente que esa parte sólo tiene que ser ejecutada si el dispositivo del usuario es mayor o igual a la API 11. La siguiente porción de código es un ejemplo de lo que los desarrolladores deben hacer:

```
if (Build.VERSION.SDK_INT > Build.VERSION_CODES.GINGERBREAD_MR1) {
    // Aquí va código superior a la API 10 de Gingerbread
}
```

Esto provoca que muchas veces el desarrollador deba programar una funcionalidad más de una vez. Actualmente muchos desarrolladores están optando por dar soporte a sus aplicaciones desde Ice Cream Sandwich en adelante, debido principalmente a la gran recepción que ha tenido Jelly Bean y a la caída constante que está teniendo Gingerbread. Si se toma en cuenta que este último sistema operativo fue lanzado el año 2010 y aún cuenta con cerca de un 20 %, se puede apreciar claramente el nivel de fragmentación que existe, principalmente por la rapidez con la que Android ha estado mejorando su sistema operativo, lanzando aproximadamente una nueva versión cada año.

A continuación, en el gráfico de la figura 2.5 se puede ver la distribución histórica de versiones que ha tenido Android con el pasar de los años. Si bien se puede ver que aún existe una gran fragmentación, esto ha ido disminuyendo y Android cada vez se está convirtiendo en un sistema operativo más estable y maduro. Por ejemplo, si se compara el porcentaje que tenía Gingerbread en el año 2013 (39.8 %) con el de este año (17.8 %) se ve que hay una diferencia sustancial, y gran parte de este porcentaje se ha trasladado a Jelly Bean, que el año pasado contaba con un 25 % del mercado y hoy en día cuenta con más del 60 %.

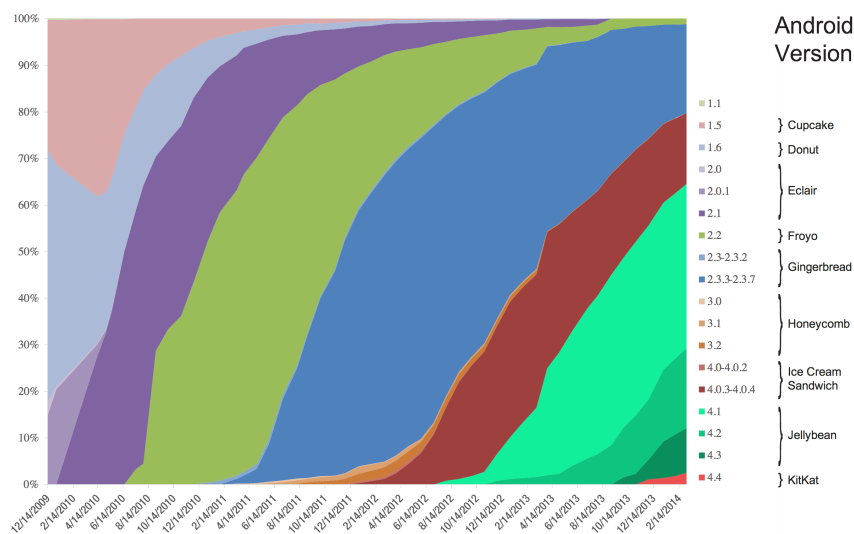


Figura 2.5: Distribución histórica de versiones de Android

Muchas veces los crashes y errores en los que la aplicación deja de funcionar correctamente, ocurren en un sistema operativo más antiguo y ya fueron arreglados en los sistemas operativos más nuevos. Esto ocasiona que al probar la aplicación en un smartphone como un Nexus 4 o como un Samsung S3 no ocurran problemas que si podrían verse en dispositivos más antiguos. Además,

debido a que a veces los operadores o fabricantes no ofrecen actualizaciones al sistema actual, el dispositivo no es actualizado, por lo que estos errores persistirán.

2.2.2. Fragmentación a nivel de hardware

Este tipo de fragmentación es la que más afecta a los desarrolladores. Por un lado, la gran cantidad de dispositivos es la que ha permitido la rápida evolución de Android, ya que cualquier fabricante puede adaptar el sistema operativo a sus necesidades e incluirlo en su hardware. Para los desarrolladores, este tipo de fragmentación es la que genera más pesadillas ya que cada dispositivo con Android cuenta con diferentes características.

Fragmentación en los tamaños de pantalla

Existen cuatro tamaños generales de pantallas: pequeña (small), normal, grande (large) y extra grande (xlarge). Para optimizar la experiencia del usuario, muchas veces se debe implementar una interfaz distinta para un tipo específico de pantalla. Esto se lleva a cabo a través de un archivo llamado *layout*, escrito en XML que definirá los elementos presentes en la interfaz.

Si se desea usar el mismo *layout* para todas las pantallas simplemente se deben ir guardando estos archivos en la carpeta de recursos del proyecto, *res/layout*. En caso que se quiera agregar un *layout* distinto para un tipo de pantalla, además de tener un archivo en la ruta antes mencionada, se debe crear una nueva carpeta de *layouts*, añadiendo el sufijo que corresponda a cada tamaño de pantalla.

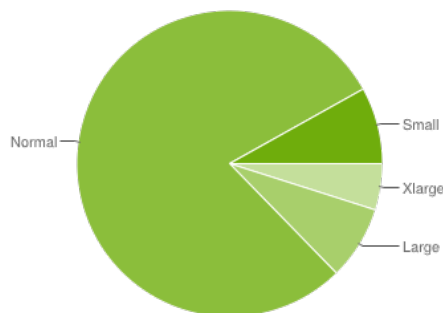


Figura 2.6: Gráfico con la distribución de tamaños de pantalla.

Por ejemplo, si se quisiera agregar un *layout* distinto para pantallas grandes (large), se debe añadir un archivo de *layout* a la carpeta *res/layout-large*, por lo que existirán dos versiones de este archivo, uno en la carpeta antes señalada,

mientras que el otro estará en la carpeta general de layouts, *res/layout*.

En el sitio web de Android [2], se entregan estadísticas mensuales sobre el porcentaje de dispositivos que tiene cada tamaño de pantalla. En el gráfico de la figura 2.6 se muestra la distribución de tamaños de pantalla durante el mes de Abril del 2014.

Fragmentación en las resoluciones de pantalla

Android categoriza las resoluciones de cada pantalla en base a la densidad de pixeles que poseen. Existen cinco tipos de densidades: baja (ldpi), media (mdpi), alta (hdpi), extra-alta (xhdpi) y extra-extra-alta (xxhdpi). Además existe otro tipo de densidad, la cual es usada principalmente para televisores (tvdpi).

Muchas veces es necesario agregar diferentes versiones del mismo recurso gráfico. Por ejemplo, si sólo se agrega un recurso gráfico en la resolución más alta (xxhdpi), este probablemente se verá bien en resoluciones altas como xxhdpi y xhdpi, pero para las más bajas, como hdpi o mdpi, la imagen será redimensionada y se perderá mucha calidad.

Al agregar elementos de UI como botones, textos, tablas, entre otros, también se debe considerar la densidad de pixeles, ya que si el ancho y alto se asignan en pixeles, va a ocurrir lo que se muestra en la figura 2.7. Por ello, es muy importante asignar valores que esten en la escala de densidad de pixeles.

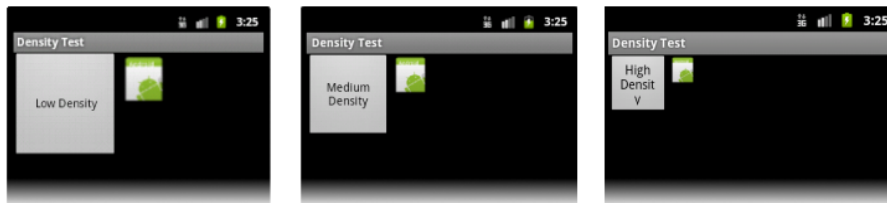


Figura 2.7: Consecuencias de asignar alto y ancho en pixeles en vez de en densidad de pixeles.

Aún con estas precauciones, es muy difícil saber cuales son los valores de alto y ancho adecuados que deben ser asignados. Esto se debe a que existe una gran diferencia entre cada tipo de resolución, ya que si por ejemplo se asignara a un elemento un ancho de 350dp, no existirán problemas en una pantalla con una resolución alta como un Nexus 4 (xhdpi), ya que el elemento tomaría un espacio de 700px, pero para un dispositivo con resolución más baja como un Samsung Galaxy Young (ldpi) si existirían, ya que el elemento tomaría 262.5px y el ancho de la pantalla de este dispositivo es de 240px. A través del siguiente sitio [?], se puede saber cuál es la densidad de pixeles de prácticamente todos los dispositivos más populares de Android.

En el gráfico de la figura 2.8 se puede apreciar la información que entrega Google sobre la distribución de densidad existente durante el mes de Abril del 2014 [2].

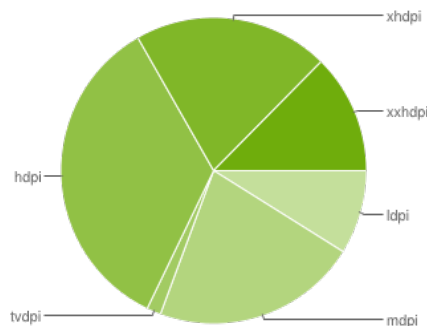


Figura 2.8: Gráfico con la distribución de densidades de pantalla.

Fragmentación en otras características

Además de los tamaños de pantalla y sus resoluciones, existen otras características muy importantes que los desarrolladores deben tener en cuenta. Una de las más importantes es la RAM con la que cuenta el dispositivo, ya que es allí donde se cargan las instrucciones que ejecuta el procesador. También es muy importante el procesador y la cantidad de núcleos con los que cuenta.

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetuer signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

2.2.3. Otros problemas

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetuer signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

Capítulo 3

Herramientas Actuales

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.1. Herramientas de Testing

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

DIVIDIR ENTRE DISTINTOS TIPOS DE TESTING

3.1.1. JUnit

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.1.2. EasyMock

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an

eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.1.3. PowerMock

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.1.4. Mockito

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.1.5. Espresso

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.1.6. Robotium

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.1.7. Robolectric

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an

eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.1.8. Spoon

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.2. Herramientas de Crashes

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.2.1. Crittercism

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.2.2. Bugsense

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.2.3. Crashlytics

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an

eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.2.4. ACRA

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

3.2.5. Google Analytics

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetur signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per.

Capítulo 4

Análisis Comparativo

Para este análisis comparativo hemos filtrado las aplicaciones que pueden ser más útiles y que pueden ser comparadas, ya que muchas de las herramientas estudiadas son muy distintas una de otra, por lo que es muy complicado realizar una comparación.

Sed iusto nihil populo an, ex pro novum homero cotidieque. Te utamur civibus eleifend qui, nam ei brute doming concludaturque, modo aliquam facilisi nec no. Vidisse maiestatis constituam eu his, esse pertinacia intellegam ius cu. Eos ei odio veniam, eu sumo altera adipisci eam, mea audiam prodesset persequeris ea. Ad vitae dictas vituperata sed, eum posse labore postulant id. Te eligendi principes dignissim sit, te vel dicant officiis repudiandae.

Id vel sensibus honestatis omittantur, vel cu nobis commune patrioque. In accusata definiebas qui, id tale malorum dolorem sed, solum clita phaedrum ne his. Eos mutat ullum forensibus ex, wisi perfecto urbanitas cu eam, no vis dicunt impetus. Assum novum in pri, vix an suavitate moderatius, id has reformidans referrentur. Elit inciderint omittantur duo ut, dicit democritum signiferumque eu est, ad suscipit delectus mandamus duo. An harum equidem maiestatis nec. At has veri feugait placerat, in semper offendit praesent his. Omnium impetus facilis sed at, ex viris tincidunt ius. Unum eirmod dignissim id quo. Sit te atomorum quaerendum neglegentur, his primis tamquam et. Eu quo quot veri alienum, ea eos nullam luptatum accusamus. Ea mel causae phaedrum reprimique, at vidisse dolores occurreret nam.

Capítulo 5

Implementación en entorno real

Sed iusto nihil populo an, ex pro novum homero cotidieque. Te utamur civibus eleifend qui, nam ei brute doming concludaturque, modo aliquam facilisi nec no. Vidisse maiestatis constituam eu his, esse pertinacia intellegam ius cu. Eos ei odio veniam, eu sumo altera adipisci eam, mea audiam prodesset persequeris ea. Ad vitae dictas vituperata sed, eum posse labore postulant id. Te eligendi principes dignissim sit, te vel dicant officiis repudiandae.

Id vel sensibus honestatis omittantur, vel cu nobis commune patrioque. In accusata definiebas qui, id tale malorum dolorem sed, solum clita phaedrum ne his. Eos mutat ullum forensibus ex, wisi perfecto urbanitas cu eam, no vis dicunt impetus. Assum novum in pri, vix an suavitate moderatius, id has reformidans referrentur. Elit inciderint omittantur duo ut, dicit democritum signiferumque eu est, ad suscipit delectus mandamus duo. An harum equidem maiestatis nec. At has veri feugait placerat, in semper offendit praesent his. Omnium impetus facilis sed at, ex viris tincidunt ius. Unum eirmod dignissim id quo. Sit te atomorum quaerendum neglegentur, his primis tamquam et. Eu quo quot veri alienum, ea eos nullam luptatum accusamus. Ea mel causae phaedrum reprimique, at vidisse dolores occurreret nam.

Capítulo 6

Conclusión

Sed iusto nihil populo an, ex pro novum homero cotidieque. Te utamur civibus eleifend qui, nam ei brute doming concludaturque, modo aliquam facilisi nec no. Vidisse maiestatis constituam eu his, esse pertinacia intellegam ius cu. Eos ei odio veniam, eu sumo altera adipisci eam, mea audiam prodesset persequeris ea. Ad vitae dictas vituperata sed, eum posse labore postulant id. Te eligendi principes dignissim sit, te vel dicant officiis repudiandae.

Id vel sensibus honestatis omittantur, vel cu nobis commune patrioque. In accusata definiebas qui, id tale malorum dolorem sed, solum clita phaedrum ne his. Eos mutat ullum forensibus ex, wisi perfecto urbanitas cu eam, no vis dicunt impetus. Assum novum in pri, vix an suavitate moderatius, id has reformidans referrentur. Elit inciderint omittantur duo ut, dicit democritum signiferumque eu est, ad suscipit delectus mandamus duo. An harum equidem maiestatis nec. At has veri feugait placerat, in semper offendit praesent his. Omnium impetus facilis sed at, ex viris tincidunt ius. Unum eirmod dignissim id quo. Sit te atomorum quaerendum neglegentur, his primis tamquam et. Eu quo quot veri alienum, ea eos nullam luptatum accusamus. Ea mel causae phaedrum reprimique, at vidisse dolores occurreret nam.

Bibliografía

- [1] Android. <http://developer.android.com/about/dashboards/index.html>. Android Dashboards.
- [2] Android. <http://www.android.com/phones-and-tablets/>. Great devices with the best of Android.

Índice de figuras

2.1. Arquitectura de Android, compuesta por cuatro capas	10
2.2. Últimos smartphones y tablets destacadas en el sitio de Android.	12
2.3. Versiones de Android.	13
2.4. Estadísticas relativas al número de dispositivos que tiene cada versión de Android en Abril del 2014.	14
2.5. Distribución historica de versiones de Android	15
2.6. Gráfico con la distribución de tamaños de pantalla.	16
2.7. Consecuencias de asignar alto y ancho en pixeles en vez de en densidad de pixeles.	17
2.8. Gráfico con la distribución de densidades de pantalla.	18