

UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS VILLAS  
FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN



**Repositorio de aplicaciones para *smartphones***

**con tecnologías Android**

Autor: Carlos Javier García Julián

Tutor: Ing. Amaury González Muro

**Santa Clara, 2014**

“Año 56 del Triunfo de la Revolución”

## Dictamen con derechos de autor para MFC



Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Ingeniería Informática, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del  
Seminario

## ***Agradecimientos***

*A mi tutor por la paciencia y la ayuda.*

*A mis profesores, principal fuente de formación de un carácter profesional.*

*A mis compañeros, fuente de apoyo y motivo para proseguir.*

*A los que están y los que ya no están, por demostrar que permanecer no es simple y llegar una ardua tarea.*

*En especial a mis padres, por la vida;*

*a mi familia, por saberme guiar hacia la luz.*

*A todos ellos les dedico el presente, fruto de más de 24 años de trabajo.*

## **RESUMEN**

El mundo de hoy respira y se alimenta de información, mantenerse al día es palabra de orden. Se hace uso de las recientes tecnologías móviles para construir un conjunto de herramientas, que cumplen con los contenidos esenciales, para la actualización de las aplicaciones de los dispositivos móviles que emplean Android como sistema operativo. Llevando como consigna que esta solución sirva como complemento de los métodos convencionales, ofrezca facilidades contribuyendo al ensanchamiento del horizonte informativo universitario y contribuya de manera alternativa a las principales tiendas de aplicaciones existentes, principal objetivo del presente trabajo.

El sistema creado es una aplicación con ambiente de fácil uso que sigue los principales estilos de diseño específicos de aplicaciones Android. El propósito del sistema implementado es ser útil en las manos de los usuarios finales y mediante este sea posible buscar, descargar e instalar aplicaciones en los dispositivos empleando para ello la conexión mediante redes inalámbricas. Es por ello que en el presente trabajo se describe el desarrollo de un conjunto de herramientas conformadas por un repositorio y una aplicación cliente capaz de consumir los servicios que el anterior ofrece.

# ABSTRAC

# INDICE

<b>INTRODUCCIÓN</b> -----	<b>1 -</b>
<b>CAPÍTULO 1 - Estudio de la factibilidad de un sistema de gestión de aplicaciones Android en la UCLV. Tiendas virtuales para <i>smartphones</i></b> -----	<b>4 -</b>
1.1. Objeto de estudio-----	4 -
1.2. Objetivos estratégicos del sistema propuesto -----	5 -
1.3. Sistemas automatizados existentes en la UCLV que funcionen como repositorio de aplicaciones Android-----	5 -
1.4. Fundamentación de los objetivos propuestos -----	6 -
1.5. Dispositivos móviles y sistema operativo Android. Tendencias y tecnologías actuales para tiendas de aplicaciones-----	8 -
1.1. El sistema operativo Android. Características y desarrollo de aplicaciones -----	9 -
1.2. Arquitectura del sistema operativo Android-----	9 -
1.3. Desarrollo de aplicaciones para Android. SDK y ADT-----	11 -
1.4. Componentes de software en el desarrollo de aplicaciones Android -----	13 -
1.5. Persistencia de datos en el desarrollo de aplicaciones Android-----	13 -
1.6. Tendencias y tecnologías actuales para tiendas virtuales de aplicaciones Android - -	14 -
1.7. Metodología del proceso de desarrollo del sistema propuesto-----	16 -
<b>CAPÍTULO 2 - Modelo del Negocio y Captura de Requisitos para la realización del trabajo</b> -	<b>18 -</b>
2.1 Modelo del negocio actual. -----	18 -
2.2 Reglas del negocio a considerar. -----	20 -
3.1 Casos de Uso del Negocio. Actores del negocio.-----	21 -
2.3 Trabajadores del negocio.-----	24 -

2.4	Actores del sistema a automatizar.-----	- 24 -
2.5	Definición de los requisitos funcionales. -----	- 25 -
2.6	Definición de los requisitos no funcionales.-----	- 25 -
2.7	Diagramas de Casos de Uso del Sistema -----	- 26 -
 <b>CAPÍTULO 3 - Herramientas, repositorio y su herramienta de manejo y aplicación cliente en Android. - 32 -</b>		
3.1	Arquitectura del Sistema -----	- 32 -
3.2	Diagrama de clases de diseño del CUS Descargar APK -----	- 33 -
3.3	Diagrama de secuencia del CUS Descargar APK -----	- 34 -
3.4	Tratamiento de errores-----	- 35 -
3.5	Diseño de la base de datos -----	- 35 -
3.6	Modelo de componentes -----	- 37 -
3.7	Diagrama de despliegue-----	- 38 -
 <b>CAPÍTULO 4 -Planificación del sistema. - 40 -</b>		
4.1.	Planificación basada en el método de estimación por Caso de Uso -----	- 40 -
4.2.	Estimación basada en Casos de Uso. -----	- 40 -
4.2.1.	Análisis de Costo – Beneficio.-----	- 47 -
<b>Conclusiones -----</b>		<b>- 49 -</b>
<b>Recomendaciones -----</b>		<b>- 50 -</b>
<b>Referencias -----</b>		<b>- 51 -</b>

## INTRODUCCIÓN

La fabricación por parte de la compañía IBM<sup>1</sup>, del primer teléfono inteligente (*smartphone* en inglés) en el año 1992, marcó el comienzo de una nueva era tecnológica, marcada por la diferencia en la interacción de los usuarios con los teléfonos móviles. Estos dispositivos inteligentes cuentan con una capacidad de conectividad, almacenamiento y procesamiento de datos que le permiten realizar actividades y procesos semejantes a una computadora personal.

Existen en la actualidad varios sistemas operativos para teléfonos inteligentes y tabletas (en inglés *tablet*, de mayor tamaño que un teléfono inteligente) y una gran cantidad de aplicaciones para cada uno de estos sistemas operativos. Estas aplicaciones pueden ser descargadas o compradas por los usuarios de los *smartphones* desde internet, explotando la capacidad de conectividad de estos, convertidos en los principales dispositivos para realizar conexiones a la red de redes.

El desarrollo de aplicaciones para *smartphones* sigue un rápido flujo validado por los cientos de aplicaciones que salen al mercado diariamente, donde el 90% de estas son ejecutadas en menos de diez ocasiones. Las compañías desarrolladoras de los sistemas operativos han desarrollado tiendas *online* donde se agrupan para su comercialización y distribución todo el cúmulo de aplicaciones existentes para su sistema operativo. Estas tiendas provén además, una aplicación cliente de la tienda que facilita la búsqueda de aplicaciones en la tienda y la instalación de estas en el dispositivo. Entre las principales tiendas virtuales existentes se encuentran la AppStore de Apple Inc., Google Play Store de Google Inc. y Windows Store de la Microsoft. Vale destacar que a pesar de que estas tiendas están destinadas a la comercialización de aplicaciones, muchas de estas se distribuyen gratuitamente.

Los usuarios cubanos poseedores de dispositivos móviles inteligentes se ven imposibilitados de acceder a estas tiendas debido a la ausencia de navegación directa a internet mediante las redes

---

<sup>1</sup>International Business Machines Corporation.

3G provistas por la compañía CubaCell y las limitantes de conectividad existentes en las instituciones que ofertan servicios de WI-FI, acotando que las redes 3G, 4G y la WI-FI constituyen las únicas vías por las cuales un *Smartphone* puede acceder a internet. Ante esta situación, estos usuarios que desean instalar aplicaciones se ven precisados de realizar el tedioso y poco efectivo proceso de buscar desde un ordenador personal las aplicaciones en internet o en repositorios de terceros y luego descargarlas, copiarlas al teléfono y luego instalarlas manualmente.

La Universidad Central “Marta Abreu de Las Villas” de las Villas (UCLV) constituye una de las instituciones educacionales que cuenta con red WI-FI en casi toda su extensión geográfica. El acceso a dicha red constituye un privilegio de la totalidad de trabajadores y estudiantes de la UCLV.

Los elementos antes descritos preceden y enuncian la realización del presente trabajo de diploma, que persigue los siguientes objetivos:

**Objetivo general:** Desarrollar las aplicaciones necesarias para brindar a los usuarios de la UCLV que cuentan con dispositivos móviles Android, de un mecanismo que les posibilite instalar aplicaciones de manera sencilla, ágil y organizada desde un sitio centralizado accesible a través de la WI-FI de la red UCLV.

De este objetivo general se desprenden los siguientes **objetivos específicos**:

- Crear un repositorio de aplicaciones Android.
- Desarrollar una aplicación que permita administrar el repositorio.
- Implementar una aplicación cliente Android que consuma la información y las aplicaciones brindadas por el repositorio.
- Desplegar el sistema para su prueba dentro de la UCLV.

En consecuencia con los objetivos planteados para el desarrollo del presente trabajo, se parte de la siguiente **hipótesis**:

La implementación de un sistema compuesto por un repositorio de aplicaciones Android, una aplicación para administrar dicho repositorio y una aplicación cliente Android que permita buscar e instalar las aplicaciones existentes en dicho repositorio puede constituir una alternativa viable a la utilización de la tienda virtual Google Play Store.

Para la presentación del contenido y resultados de la investigación e implementación del sistema se utiliza el presente informe de Tesis de Pregrado cuyos objetivos se estructuran de la siguiente forma:

- un primer capítulo donde se establece un marco teórico referencial desarrollado a partir de la consulta de bibliografía actualizada sobre los temas que constituyen la plataforma de implementación del sistema,
- un segundo capítulo que presenta a través del modelo de Negocio y los requisitos las características de la situación que sugiere el desarrollo del sistema,
- un tercer capítulo exponiendo la descripción de la propuesta de solución que se materializa en el sistema desarrollado,
- un cuarto capítulo donde se exponen las pruebas y el análisis de factibilidad del sistema propuesto,
- un acápite de conclusiones y uno de recomendaciones donde se exponen la forma en que se lograron los objetivos propuestos, así como las pautas a tener en cuenta para futuras implementaciones del sistema y
- finalmente un acápite de bibliografía que relaciona los documentos consultados para el desarrollo de la presente investigación.

## **CAPÍTULO 1 - Estudio de la factibilidad de un sistema de gestión de aplicaciones Android en la UCLV. Tiendas virtuales para *smartphones***

Las tiendas de aplicaciones para dispositivos móviles han cobrado una gran popularidad en la actualidad, permitiendo a los usuarios obtener, mediante pago o gratuitamente, aplicaciones capaces de realizar diversas tareas y solucionar problemas gracias a las características funcionales. En el presente capítulo se abordan los fundamentos, la factibilidad y los requerimientos tecnológicos estudiados que nos permiten enunciar la propuesta de la implementación de un sistema que posibilite, a los usuarios de la UCLV que cuentan con dispositivos móviles Android, instalar aplicaciones de manera sencilla, ágil y organizada desde un sitio centralizado accesible a través de la WI-FI de la red. Este sistema en el presente trabajo es llamado a partir de este momento como *Jaas*, siglas del inglés “Just Another Android Store” cuya traducción al español es “solamente otra tienda Android”.

### **1.1. Objeto de estudio**

El desarrollo de aplicaciones para *smartphones* sigue un rápido flujo validado por los cientos de aplicaciones que salen al mercado diariamente. El cúmulo de aplicaciones resulta tan elevado que para facilitar la búsqueda por los usuarios estas se han agrupado en categorías que abarcan desde la comunicación, educación, fotografía, entretenimiento, compras, música y video entre otras. Las principales compañías productoras de sistemas operativos para *smartphones* tales como AppStore Apple Inc., Google Inc. y Microsoft han desarrollado tiendas *online* donde se agrupan para su comercialización y distribución todo el cúmulo de aplicaciones existentes para el sistema operativo que estas producen. Estas tiendas provén además, una aplicación cliente de la tienda que facilita la búsqueda de aplicaciones en la tienda y la instalación de estas en el dispositivo móvil. Entre las principales tiendas virtuales existentes se encuentran la AppStore de Apple Inc. para el sistema operativo iOS, Google Play Store de Google Inc. para Android y

Windows Store de la Microsoft para WindowsPhone. Vale destacar que a pesar de que estas tiendas están destinadas a la comercialización de aplicaciones, muchas de estas se distribuyen gratuitamente.

## **1.2. Objetivos estratégicos del sistema propuesto**

Partiendo de las premisas de que el sistema operativo Android resulta el más popular entre los SO existentes (Ver) y que la UCLV cuenta con red WI-FI en casi toda su extensión geográfica y donde el acceso a dicha red constituye un privilegio de la totalidad de trabajadores y estudiantes, se propone la implementación de un conjunto de herramientas para facilitar la instalación de aplicaciones de dispositivos móviles con tecnología Android desde un sitio centralizado accesible a través de la WI-FI de la red UCLV. El sistema, conformado por una aplicación cliente y otra de administración del servidor, se presenta al usuario cliente de forma amena y proveyendo facilidades no acostumbradas debido a la inexistencia de sistemas similares en la UCLV.

## **1.3. Sistemas automatizados existentes en la UCLV que funcionen como repositorio de aplicaciones Android**

En la UCLV no existe en la actualidad un mecanismo automatizado que permita a los usuarios de dispositivos Android instalar aplicaciones de manera sencilla, ágil y organizada desde un sitio centralizado accesible a través de la WI-FI de la red UCLV. Estos usuarios que desean instalar aplicaciones se ven precisados de realizar el tedioso y poco efectivo proceso de buscar desde un ordenador personal las aplicaciones en internet o en el servidor samba de archivos compartidos de la UCLV con dirección \\10.12.1.64\Softlib\AndroidApks y luego descargarlas, copiarlas al teléfono y luego instalarlas manualmente.

#### **1.4. Fundamentación de los objetivos propuestos**

Las compañías desarrolladoras de los sistemas operativos han desarrollado tiendas *online* donde se agrupan para su comercialización y distribución todo el cúmulo de aplicaciones existentes para su sistema operativo. Estas tiendas provén además, una aplicación cliente de la tienda que facilita la búsqueda de aplicaciones en la tienda y la instalación de estas en el dispositivo

Los usuarios cubanos poseedores de dispositivos móviles inteligentes se ven imposibilitados de acceder a estas tiendas debido a la ausencia de navegación directa a internet mediante las redes 3G provistas por la compañía CubaCell y las limitantes de conectividad existentes en las instituciones que ofertan servicios de WI-FI, acotando que las redes 3G, 4G y la WI-FI constituyen las únicas vías por las cuales un *Smartphone* puede acceder a internet. Ante esta situación, estos usuarios que desean instalar aplicaciones se ven precisados de realizar el tedioso y poco efectivo proceso de buscar desde un ordenador personal las aplicaciones en internet o en repositorios de terceros y luego descargarlas, copiarlas al teléfono e luego instalarlas manualmente.

Desarrollar las aplicaciones necesarias para brindar a los usuarios de la UCLV que cuentan con dispositivos móviles Android, de un mecanismo que les posibilite instalar aplicaciones de manera sencilla, ágil y organizada desde un sitio centralizado accesible a través de la WI-FI de la red UCLV.

El desarrollo de *Jaas* parte de la necesidad de brindar soporte dinámico de aplicaciones a los *smartphones* Android, aprovechando las bondades y disímiles formas de conectividad que brindan estos dispositivos. La elección del SO al cual darle soporte de aplicaciones está basado en el estudio de las estadísticas que reflejan que en la actualidad, a pesar de que existen una variedad de sistemas operativos (SO), como Android, iOS, Windows Phone, FirefoxOs, Ubuntu Phone...etc.; son los dispositivos Android los que ocupan el 75% de los smartphones en el mundo debido a que entre otras razones resultan los más baratos en el mercado. Por estos

motivos se fundamenta el desarrollo de un sistema que brinde aplicaciones compatibles con plataforma Android por lo cual el cliente *Jass* se ejecuta sobre *smartphones* con este SO.

Es en este segmento donde se manifiesta el primero de los objetivos específicos definidos en la introducción del presente trabajo, consistente en la creación de un repositorio de aplicaciones Android. La realización de este repositorio viabiliza el consumo de los servicios que brinda por una aplicación Android cliente.

El segundo de los objetivos se centra en desarrollar una aplicación que permita administrar el repositorio, consistiendo en una herramienta escrita en PHP con comandos básicos de la consola de Linux preparada para generar, actualizar, limpiar y descargar descripciones de Google Play Store que esta brinda gratuitamente. Esta herramienta o aplicación ejecutable mediante la consola de Linux, tiene como objetivo establecer un procedimiento modelo en el consumo de sus servicios que permita explorar sus aplicaciones por parte de los usuarios incluso encontrándose estos últimos offline. Esta herramienta será denominada a partir de este momento como *repoGen*, alegórico a "*Repository Generator*".

El tercer objetivo contempla la creación de la aplicación cliente *Jaas* la cual se encargue de descargar aplicaciones del repositorio y luego instalarlas de forma que este proceso permanezca transparente al usuario cuando seleccione una aplicación para instalarla. *Jaas* posibilita además la visualización de la información contemplada en el repositorio de aplicaciones de manera offline de manera que el usuario no necesita tener conexión WI-FI para navegar por las listas de aplicaciones disponibles, no siendo así para instalarlas, proceso que requiere la conexión al servidor de aplicaciones. A través de *Jass* el usuario puede realizar búsquedas de aplicaciones las cuales pueden ser agrupadas por categorías de forma dinámica y al ser seleccionados se visualiza su información textual o descripción. Toda esta información puede ser actualizada consumiendo del repositorio la versión actualizada de sus bases de datos.

Desplegar el sistema para su prueba y posterior empleo dentro de la UCLV constata para este trabajo una piedra angular, por la importancia que trae como consecuencia una solución

innovadora en el campo. Es por ello que esto conforma el cuarto y final de los objetivos específicos en los cuales se fundamenta el presente trabajo de diploma. No obstante a este sistema se le aplican pruebas de caja negra para chequear el comportamiento del sistema frente a valores extremos con el fin de lograr una primera versión totalmente funcional. Además la opinión pública de los usuarios de *Jaas* se registra para la obtención de futuras versiones. Así como estadísticas del uso de la misma registrando las descargas por los usuarios.

### **1.5. Dispositivos móviles y sistema operativo Android. Tendencias y tecnologías actuales para tiendas de aplicaciones**

Para comenzar con la fundamentación teórica del desarrollo del sistema propuesto resulta indispensable repasar el concepto y la historia de los smartphones y del sistema operativo Android. En el año 1992, la compañía IBM fabricó el IBM Simon, reconocido como el primer teléfono inteligente, fue distribuido por EE.UU entre agosto de 1994 y febrero del 1995, a un precio de \$899, con pantalla táctil LCD monocromo y una interfaz de usuario con pocos botones físicos. Estos dispositivos en su primera generación tenían como objetivo a los grandes empresarios, principalmente por incluir muchas aplicaciones útiles, como una libreta de direcciones, calendario con citas, agenda, calculadora, reloj mundial, bloc de notas electrónico y anotaciones manuscritas a mano alzada, y además el alto precio exhibido. Otro de los saltos importantes en este aspecto fue la aparición en 2007 del iPhone, de la mano de Apple, esto revoluciona la forma de ver estos teléfonos hasta la fecha, tildados de poco cómodos por su tamaño, y además muy complejos. El iPhone con un sistema bastante intuitivo y con dimensiones minúsculas con respecto a sus primos, por primera vez se hablaba de los Smartphone como el teléfono del futuro. En 2008, Google sacó a la luz un sistema operativo para Smartphone de código abierto Android.

Android es un sistema operativo basado en Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, inicialmente desarrollado por Android Inc. Google respaldó económicamente y más tarde compró esta empresa en 2005.

Android fue presentado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008.

### **1.1. El sistema operativo Android. Características y desarrollo de aplicaciones**

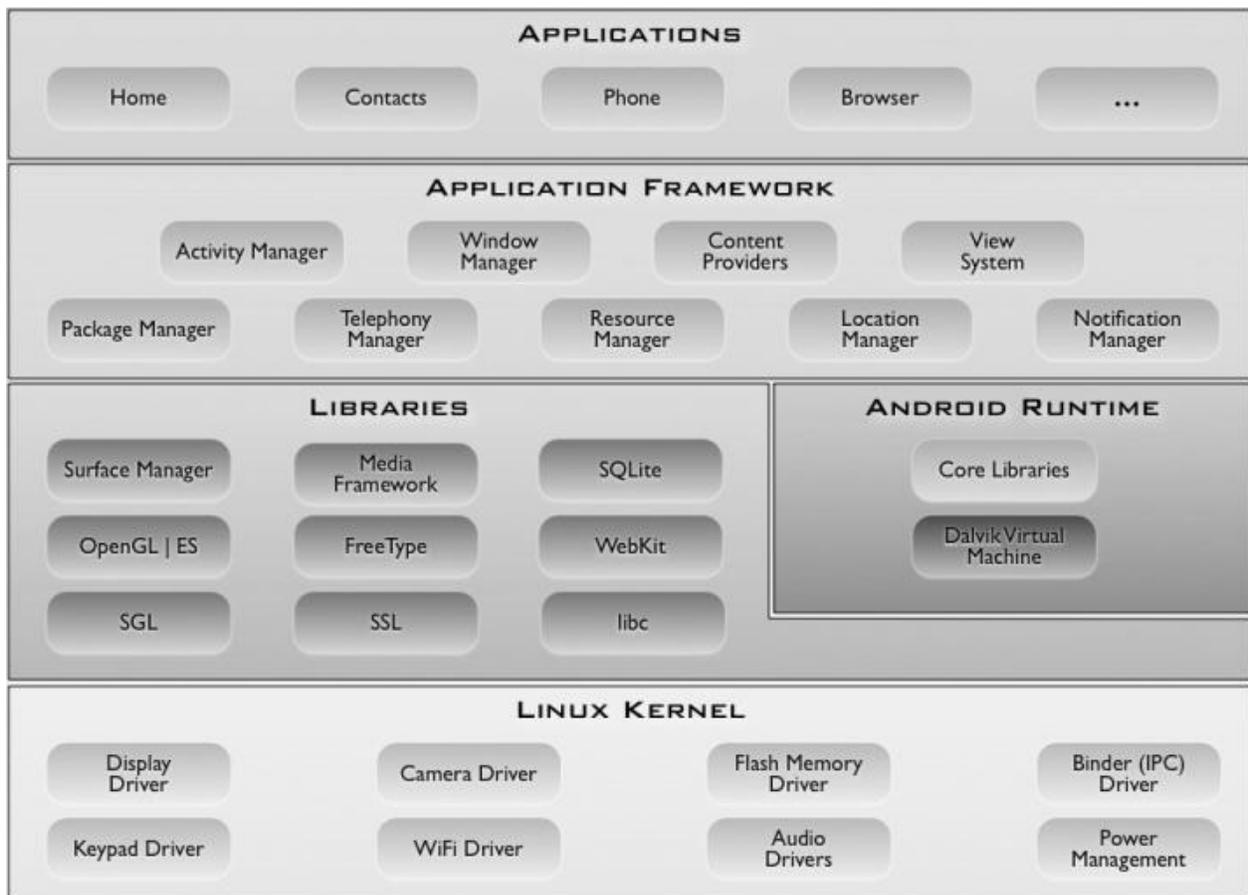
El sistema operativo Android constituye el sistema operativo más utilizado por los dispositivos inteligentes. En estadísticas de StrategyAnalytics el 79% de los smartphones vendidos durante el año 2013 incorporaban alguna versión del sistema operativo de Google, lo que supone que 4 de cada 5 smartphones vendidos en todo el mundo. Estos números nos hacen identificar a Android como el equivalente de Windows de los smartphones, pues su dominio es aplastante en la actualidad. En los siguientes apartados describimos algunas características esenciales del SO Android así como del desarrollo de aplicaciones para esta plataforma.

### **1.2. Arquitectura del sistema operativo Android**

La arquitectura del SO Android, como se muestra en la Fig. 3.1, está diseñada por varias capas que facilitan al desarrollador la creación de aplicaciones.

Sobre una capa base se encuentra un kernel o núcleo Linux 2.6 y los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software. En una segunda capa encontramos las Bibliotecas y el Entorno de ejecución de Android. En las Bibliotecas se incluyen un conjunto de librerías C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras. El Entorno de ejecución de Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada

aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato DalvikExecutable (.dex), el cual está optimizado para un consumo mínimo de memoria. La siguiente capa Marco de trabajo de aplicaciones, está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual Dalvik. En la última capa llamada Capa de aplicaciones, se incluyen todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario como las que no, las nativas (programadas en C o C++) y las administradas (programadas en Java), las que vienen preinstaladas en el dispositivo y aquellas que el usuario ha instalado.



### **1.3. Desarrollo de aplicaciones para Android. SDK y ADT**

El **SDK** ( Software Development Kit ) de Android, incluye un conjunto de herramientas de desarrollo. Comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU, documentación, ejemplos de código y tutoriales. Las plataformas de desarrollo soportadas incluyen Linux (cualquier distribución moderna), Mac OS X 10.4.9 o posterior, y Windows XP o posterior. La plataforma integral de desarrollo (IDE, Integrated Development Environment) soportada oficialmente es Eclipse junto con el complemento ADT ( Android Development Tools plugin ), aunque también puede utilizarse un editor de texto para escribir ficheros Java y XML y utilizar comandos en un terminal ( se necesitan los paquetes JDK, Java Development Kit y Apache Ant) para crear y depurar aplicaciones. Además, pueden controlarse dispositivos Android que estén conectados (ej. reiniciarlos, instalar aplicaciones en remoto).

Las Actualizaciones del SDK están coordinadas con el desarrollo general de Android. El SDK soporta también versiones antiguas de Android, por si los programadores necesitan instalar aplicaciones en dispositivos ya obsoletos o más antiguos. Las herramientas de desarrollo son componentes descargables, de modo que una vez instalada la última versión, pueden instalarse versiones anteriores y hacer pruebas de compatibilidad.

Una aplicación Android está compuesta por un conjunto de ficheros empaquetados en formato .apk y guardada en el directorio /data/app del sistema operativo Android ( este directorio necesita permisos de superusuario , root, por razones de seguridad ). Un paquete APK incluye ficheros .dex ( ejecutables Dalvik, un código intermedio compilado ), recursos, etc.

A la utilización del SDK por el desarrollador se unido al **ADT** (Android Developer Tools), el cual es un plugin que proporciona un conjunto de herramientas que se integran con el IDE Eclipse. ADT ofrece acceso a muchas características que le ayudan a desarrollar aplicaciones de Android rápidamente. ADT ofrece acceso GUI a muchas de las herramientas del SDK de línea de comandos, así como una herramienta de diseño de interfaz de usuario para la creación rápida de prototipos, diseño y construcción de la interfaz de usuario de la aplicación.

Los proyectos de aplicaciones que se desarrollan utilizando ADT y el SDK de Android, presentan la siguiente composición:

1. *src*: reservada para los paquetes contenedores de las clases del sistema.
2. *gen*: contiene los archivos java generados por el ADT, como la clase R, controladora de los recursos de las aplicaciones, y los propios recursos una vez barrida.
3. Las siguientes, *Android x.x* y *Android Private Libraries*, están reservadas para el soporte que ofrece la versión en cuestión.
4. *assets*: tiene los recursos no propios de la aplicación, los que serán usados en el tiempo de desarrollador. Los recursos que desean formar parte de la compilación final del producto, se deben guardar en una carpeta especial, no creada pero si registrada bajo el nombre raw en directorio: *res/raw/<<recurso>>*.
5. *bin*: guarda la compilación final del proyecto, esto incluye el .apk, ejecutable y producto final.
6. *libs*: aparece por defecto con la biblioteca del soporte inherente a la versión, y presta a la incorporación de nuevas bibliotecas.
7. *res*: muestra una estructura en la que confluyen los recursos de la como son las imágenes<sup>2</sup>, guardadas bajo distintas resoluciones, las vistas<sup>3</sup>, definidas a través del lenguaje de marcado XML como los menús<sup>4</sup> y los distintos valores<sup>5</sup> como las dimensiones, estilos, y recursos.
8. *AndroidManifest*: es el archivo más importante, en él se definen las actividades, recursos externos consultados por la aplicación, así como permisos a solicitar. Dependiendo de su

---

<sup>2</sup> *drawable*

<sup>3</sup> *layout*

<sup>4</sup> *menu*

<sup>5</sup> *value*

composición se generan los archivos del APK, entre ellos la información que aparece antes de la instalación.

#### **1.4. Componentes de software en el desarrollo de aplicaciones Android**

Como componentes de software importantes en la arquitectura Android se encuentran las actividades (*activities*), las vistas (*view*) y los *intents*.

Las **actividades** representan el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una actividad como el elemento análogo a una ventana en cualquier otro lenguaje visual (Sgoliver).

Los objetos *view* son los componentes básicos con los que se construye la interfaz gráfica de la aplicación, análoga por ejemplo a los controles de Java o .NET. De inicio, Android pone a disposición de los desarrolladores una gran cantidad de controles básicos, como cuadros de texto, botones, listas desplegables o imágenes, aunque también existe la posibilidad de extender la funcionalidad de estos controles básicos o crear nuestros propios controles personalizados.

Un *intent* es el elemento básico de comunicación entre los distintos componentes Android que hemos descrito anteriormente. Se pueden entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones.

Mediante un *intent* se puede mostrar una actividad desde cualquier otra, iniciar un servicio, enviar un mensaje *broadcast*, iniciar otra aplicación, etc.

#### **1.5. Persistencia de datos en el desarrollo de aplicaciones Android**

La plataforma Android proporciona dos mecanismos principales para el almacenamiento y consulta de datos estructurados: Bases de Datos SQLite y Content Providers. Por ser el principal mecanismo que se utiliza en el sistema propuesto para la persistencia de los datos de los centros e instituciones de la UCLV, se describe en este apartado el mecanismo del SQLite.

SQLite es un motor de bases de datos muy popular en la actualidad por ser de pequeño tamaño, no necesitar servidor, precisar poca configuración, ser transaccional y por supuesto por ser de código libre. Android incorpora todas las herramientas necesarias para la creación y gestión de bases de datos SQLite, y entre ellas una completa API para llevar a cabo de manera sencilla todas las tareas necesarias.

En Android, la forma típica para crear, actualizar, y conectar con una base de datos SQLite se efectúa a través de una clase auxiliar llamada *SQLiteOpenHelper*, o para ser más exactos, de una clase propia que derive de ella y que debemos personalizar para adaptarnos a las necesidades concretas de nuestra aplicación.

## **1.6. Tendencias y tecnologías actuales para tiendas virtuales de aplicaciones Android**

**Google Play** (antes **Android Market**) constituye la principal tienda de software en línea para los dispositivos con sistema operativo [Android](#). Fue desarrollada por [Google](#) y anunciada por este el 28 de agosto de 2008 poniéndola a disposición de los usuarios el 22 de octubre de 2008. Es una aplicación que está preinstalada en la mayoría de los dispositivos Android y que permite a los usuarios buscar, obtener información y descargar aplicaciones publicadas por desarrolladores terceros. En octubre de 2012, Google Play disponía de más de 700.000 aplicaciones

Las aplicaciones Android disponibles en **Google Play** pueden ser gratuitas o de pago. Inicialmente, las aplicaciones tenían una función puramente recreativa. Sin embargo, han ido evolucionando en aplicaciones más útiles, como son las aplicaciones para el registro de gastos, información deportiva, guías de restaurantes, callejeros... Actualmente las nuevas aplicaciones más innovadoras son las llamadas de [realidad aumentada](#) que combinan elementos reales y virtuales.

El tipo de aplicaciones Android se clasifican de la siguiente manera: Compras, comunicación, cómics, deportes, estilo de vida, finanzas, herramientas, multimedia, noticias y meteorología, ocio, productividad, referencia, salud, sociedad, temas, viajes, demostración y bibliotecas de software. Unas de las aplicaciones más importantes para dispositivos móviles son los juegos, y estos reciben un trato especial dentro de las aplicaciones su clasificación es la siguiente: Arcade y acción, casuales, juegos de cartas, casino, puzzles y juegos para ejercitar la mente.

Existen en el mercado tiendas virtuales Android alternativas a Google Play y muchas cuentan con cientos de miles de descargas y un funcionamiento digno de destacar. Ya sea para encontrar aplicaciones independientes que no están en **Google Play** por determinadas cuestiones o en aquellas circunstancias donde el “store” oficial no se encuentra disponible. Entre estas tiendas tenemos:

- **Amazons Apps:** La tienda que más peso logra imponer sobre Google Play es la de la gigantesca empresa Amazon. Su disponibilidad es gratuita y cuenta con un extenso catálogo de aplicaciones, juegos y libros, teniendo incluso algunas opciones que son exclusivas. Además de contar con un diseño impecable, tiene el atractivo de poner a diario una aplicación de pago para su descarga gratis.
- **Aptoid:** Cuenta con casi 200.000 aplicaciones distintas, las cuales se encuentran dentro del marco de la legalidad. Asimismo, se encuentra en las etapas previas a actualizar completamente su diseño para otorgarle un importante salto de calidad y ya ha adicionado un propio sistema de seguridad a la hora de descargar opciones que no generen un peligro para el teléfono.
- **F-Droid:** Esta es una tienda con más de 700 aplicaciones gratuitas disponibles que se destacan por contar con una licencia libre, es decir que su código fuente es accesible y puede editarse para adaptar la aplicación a las necesidades propias. Incluso, su uso no requiere poseer una cuenta de usuario, por lo que el sistema garantiza el anonimato del usuario, ya que no guarda un historial de las aplicaciones descargadas.

- **AndroidPit:** Es una tienda que suele llegar instalada de fábrica en muchos dispositivos de origen chino. Entre sus aspectos positivos, cuenta con recomendaciones, reviews propias y un buscador integrado muy completo. En su catálogo cuenta con aplicaciones tanto gratuitas como de pago, siendo estas últimas generalmente un poco más caras que en Google Play.

Además de las mencionadas anteriormente destacan las tiendas de aplicaciones SlideMe, 1Mobile y BlackmartAlpha con características similares a las anteriores pero un tanto menos populares.

## **1.7. Metodología del proceso de desarrollo del sistema propuesto**

El proceso de desarrollo del sistema a implementar, se define siguiendo AUP. La elección de esta metodología está determinada normalmente por el estudio de la cantidad de personas involucradas, los medios y el tiempo disponible. Confluyen también factores como la dependencia a la interacción con distintas instituciones de la universidad y la premura de los hechos.

El AUP<sup>6</sup> es una metodología de desarrollo ágil proveniente de RUP<sup>7</sup>, pero con particularidades notables a la hora del detallado de documentos, tomando solamente lo necesario o indispensable.

El ciclo de vida de AUP está compactado en cuatro fases:

- la **Iniciación**, la cual indica el alcance del proyecto, la arquitectura a utilizar, la implicación del personal, los medios disponibles y financiamiento,
- la **Elaboración**, que mejora la arquitectura planteada,
- la **Construcción**, donde se elabora el software funcional,

---

<sup>6</sup> *Agile Unified Process*

<sup>7</sup> *Rational Unified Process.*

- y por último la Transición, encargada de validar y desplegar el producto final.

Su naturaleza iterativa incremental se desglosa en cinco disciplinas:

- el Modelado, aborda el negocio de la organización, el problema del dominio e identifica las soluciones viables,
- la Implementación, el cual transforma el modelo en código ejecutable y lleva a cabo un nivel básico de pruebas,
- el Despliegue, donde se planifica la entrega del proyecto y lo deja en manos del usuario final, listo para usarse,
- la Administración de la Configuración, que chequea y administra las versiones y cambios del producto,
- el Entorno, encargado de gestionar el ambiente de trabajo, provee las guías y herramientas pertinentes para el desarrollo satisfactorio del grupo.(Ambler, 2006)

## **CAPÍTULO 2 - Modelo del Negocio y Captura de Requisitos para la realización del trabajo**

El siguiente capítulo se describe los mecanismos disponibles a los usuarios de las redes de la UCLV propietarios de dispositivos con tecnologías Android en su intento de descargar e instalar aplicaciones. El estudio de estos mecanismos a través de la modelación del negocio constituye una vía de introducción a la solución del problema gracias a las facilidades que brinda la modelación utilizando el Lenguaje Unificado de Modelación (UML).

### **2.1 Modelo del negocio actual.**

Una correcta modelación destaca puntos clave sobre el negocio y revela e identifica segmentos claves en los procesos que se desarrollan en el transcurso del negocio, para ello destaca como una buena forma de representar el transcurso del negocio actual es mediante un diagrama de actividades, expuesto en la Figura 2.1 Diagrama de actividades donde se representa el modelo del negocio actual. y posteriormente explicado las situaciones paso a paso en las que incurren los usuarios para buscar, descargar e instalar aplicaciones en sus dispositivos.

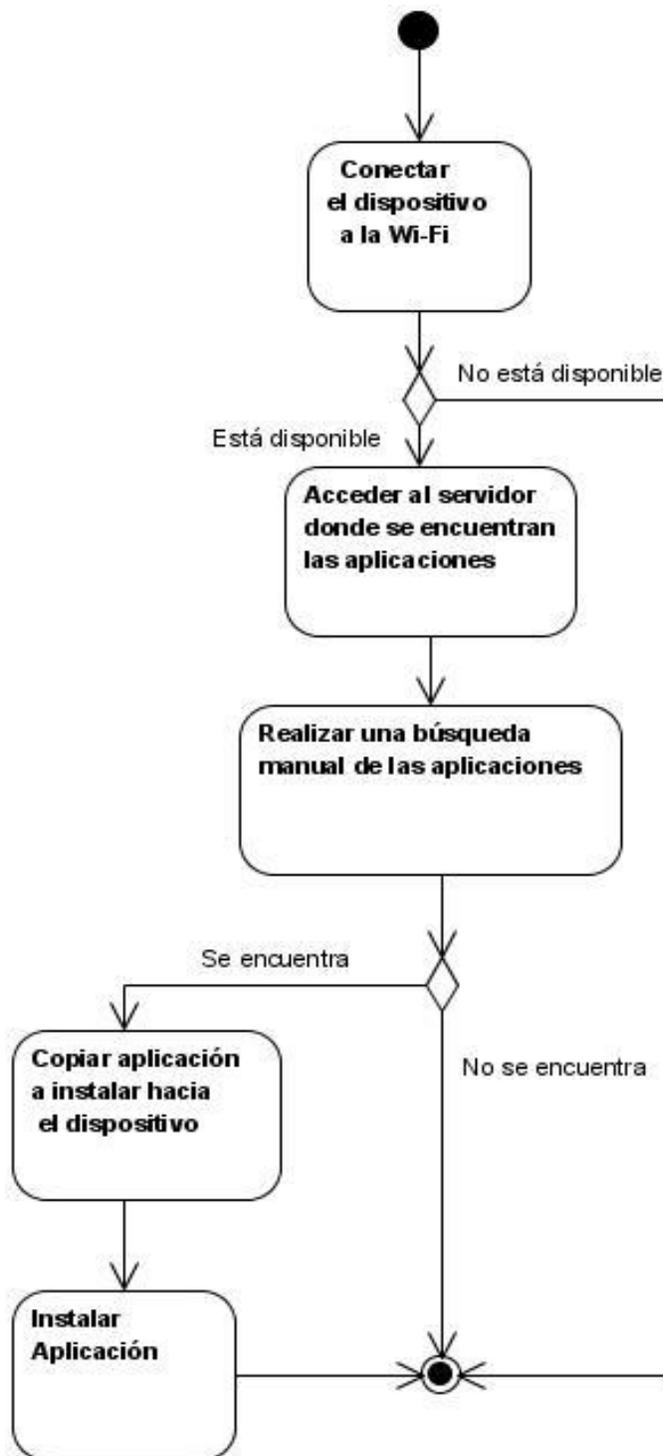


Figura 2.1 Diagrama de actividades donde se representa el modelo del negocio actual.

Para la realización de este diagrama de actividad se tiene como pre-condición que el usuario cuenta con un dispositivo para la conexión a la red (teléfono inteligente, tableta...) mediante wifi, registrado para su uso en la red de la UCLV, según las políticas de seguridad definidas. Para comenzar la ejecución del diagrama el usuario realiza la conexión del dispositivo a la red. Luego se debe acceder al servidor identificado con la dirección IP 10.12.1.64 donde se encuentra situado, entre otras muchas cosas, un pequeño repositorio de aplicaciones para Android. Seguidamente el usuario debe realizar una búsqueda manual para localizar la ubicación de la aplicación, punto crítico este del negocio, que desea; en caso de cumplir su objetivo entonces copia hacia el dispositivo la aplicación encontrada. Posteriormente en algún momento el usuario realiza su instalación en el dispositivo. Aunque siempre se desea el éxito cuando se realiza una búsqueda en un repositorio, muchas veces no es logrado el objetivo por no encontrar la aplicación o por la no existencia de la misma, en dicho caso termina el proceso con un usuario descontento y anhelante de una carente aplicación.

Es para ello que se realiza una captura de requisitos del negocio para develar, entre otras cosas, reglas del negocio inviolables en toda ocasión.

## **2.2 Reglas del negocio a considerar.**

Las políticas actuales de la UCLV, así como del departamento de redes de la misma institución son de suma importancia a tener en cuenta para la construcción, implementación e implantación de un mecanismo capaz de proporcionar a los usuarios de las redes de la UCLV las aplicaciones deseadas. Por los dichos motivos se realiza una pesquisa de requisitos la cual denota las principales acciones a automatizar y las reglas del negocio a considerar en todo momento, estas son:

- Los propietarios de dispositivos móviles deben registrar estos para poder conectarse a las redes Wi-Fi de la UCLV mediante ellos.
- Todos los usuarios pertenecientes al dominio uclv.edu.cu y uclv.cu tienen acceso al servidor de aplicaciones.

- Las aplicaciones deben estar divididas por categorías.

### 3.1 Casos de Uso del Negocio. Actores del negocio.

Un usuario o cliente en el negocio actual tiene las posibilidades de instalar y actualizar una aplicación que para ello necesita primeramente buscar y descargar de forma manual como se representa en la Figura 2.1 y detallado más adelante en la Tabla 2.1, Tabla 2.2 y Tabla 2.3. Dicho servidor es también actualizado de forma manual por los administradores encargados al mismo tiempo de organizar de la mejor forma posible.

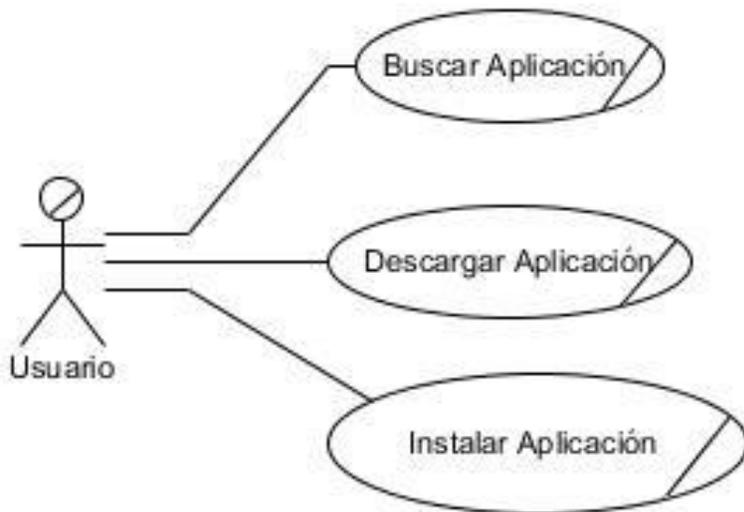


Figura 2.2 Diagrama de CU del negocio existente para la descarga de aplicaciones en la UCLV.

Caso de Uso del negocio #1	
Nombre	Buscar Aplicación

Descripción	Como su nombre da a conocer la base de este proceso del negocio se realiza una búsqueda manual en el repositorio de la aplicación deseada por parte del usuario.
Actores asociado	Usuario

*Tabla 2.1 Caso de Uso del Negocio Buscar Aplicación.*

Caso de Uso del negocio #2	
Nombre	Descargar Aplicación
Descripción	Este caso de uso en el negocio no es lo que comúnmente se conoce como descarga, sino copiar una aplicación del servidor y pegarla en la memoria del dispositivo, todo esto de forma manual. Para ello es requerido la instalación de un navegador de archivos, de otra manera es imposible realizar esta operación directamente del dispositivo.
Actores asociado	Usuario

*Tabla 2.2 Caso de Uso del Negocio Descargar Aplicación.*

---

Caso de Uso del negocio #3	
Nombre	Instalar Aplicación
Descripción	Al igual que el caso de uso Descargar Aplicación es imprescindible la presencia de un navegador de archivos. En este proceso se debe buscar el APK ( <i>Android Application Package File</i> ) descargado con anterioridad y ejecutarlo manualmente.
Actores asociado	Usuarios

**Tabla 2.3 Caso de Uso del Negocio Instalar Aplicación.**

En el caso de uso presentado en la Figura 2.2 se tiene como actor del negocio al usuario de la red UCLV y aunque parezca simple el proceso suele complicarse más de lo que muchas veces se quiere. Inicialmente se parte que para acceder al servidor y constatar que es un “usuario” el actor de los mencionados casos de uso del negocio debe estar conectado a las redes inalámbricas y autenticado o también es posible la conexión desde los ordenadores que dispone la universidad en sus laboratorios, en este caso también estaría autenticado por lo que es omitido el proceso de autenticación para acceder al servidor.

Similar tarea ocurre cuando el usuario desea actualizar cierta aplicación pero esta vez aún peor debido que una vez descargue y ya se tenga en el *smartphone* el apk (Paquete de aplicación de Android) es que puede corroborar si es una versión más reciente, una anterior o la misma.

Estos casos de usos son posible gracias a trabajadores del negocio, estos pueden ser personas o sistemas, que representan roles del negocio imprescindibles para el funcionamiento.

### **2.3 Trabajadores del negocio.**

Para mantener en marcha los servidores de la UCLV, monitorear las transferencias de archivos y velar por el buen funcionamiento de las redes informáticas, existen los administradores de red o comúnmente llamados como “admins”. Estos trabajadores actualizan los servidores de datos entre otros y conforman el único actor del sistema que en este negocio es intangible para el usuario; debido a que es aquel que se encarga de actualizar anónimamente el servidor y publicar nuevas aplicaciones en el repositorio. Estos trabajadores del negocio se convierten más tarde en actores del sistema.

### **2.4 Actores del sistema a automatizar.**

Es imprescindible para un sistema informático de plataformas móviles el ser útil, práctico, de fácil utilización y amigable a la vista del usuario. Por ello el sistema a automatizar está dividido en dos componentes cuenta con dos casos de uso del sistema, representando cada uno un sistema a implementar, y sendos actores dispuestos de manera central en sus subsistemas.

Para la aplicación pública o cliente se tiene Cliente como actor del sistema donde interactúa con el sistema mediante funcionalidades como “Buscar Aplicación”, “Instalar Aplicación” y “Configurar”. Mientras en otro lado el Administrador cumple con el rol de actor del sistema de administración donde interactúa con el sistema con el único objetivo de “Actualizar Repositorio” utilizado por los clientes.

Una vez definido los actores del sistema es necesario definir requisitos funcionales y no funcionales. Los requisitos funcionales son aquellos puntos fuertes en el sistema, puntos inviolables y a los que está orientado el sistema, mientras que los requisitos no funcionales se definen como necesidades de ejecución del sistema, lo que es necesario para que el sistema funcione.

## **2.5 Definición de los requisitos funcionales.**

Los requisitos funcionales del sistema están orientados hacia la búsqueda, descarga e instalación de aplicaciones y paquetes de instalación, pero además la configuración del sistema para un mejor funcionamiento. La búsqueda, lo que está denominado búsqueda en el caso de uso del sistema, no solamente es realizada a través del nombre de una aplicación, además de esto existe la búsqueda por categorías o filtrado por categorías. Es en general la acción de buscar mediante cualquiera de las diferentes variantes propuestas en el sistema, ya sea por tamaños, categorías, nombre, ordenar alfabéticamente y búsqueda de forma manual recorriendo la lista de aplicaciones. La instalación de una aplicación de aplicaciones está dividido en tres partes o tiempos, la primera que extiende de la búsqueda pues comúnmente el usuario inicia la instalación luego de realizar una búsqueda, una segunda parte que es la descarga y una vez descargada la aplicación se ejecuta para instalar. El proceso de descargar puede ser descrito como copiar desde el repositorio hacia un directorio específico en la memoria externa del dispositivo o SDCard. Es imprescindible que cada usuario pueda configurar la aplicación para utilizarla según sus características y es por ello que el hecho de configurar se define como requisito funcional del sistema.

## **2.6 Definición de los requisitos no funcionales.**

Para el despliegue del sistema es necesario tener en cuenta que la herramienta cliente debe ser ejecutada en un dispositivo con una plataforma Android versión 2.1 o superior, aunque el funcionamiento en las versiones 4.x hace un mejor uso del actionBar o barra de acción. El uso y trabajo constante con la barra de acción se estandariza en Android a partir de la versión 4.0 o “API nivel 17” y no está disponible en ediciones anteriores, pero en la herramienta se utiliza una librería denominada “Action Bar Sherlock” que permite entre otras cosas, el uso de la barra de acción en versiones anteriores a 4. Una memoria extraíble en el dispositivo con más de 3 Mbyte de espacio libre para el almacenamiento de la base de datos de aplicaciones offline y espacio adicional según las aplicaciones que se descarguen.

Por otro lado en el servidor de aplicaciones o repositorio son imprescindibles ciertos componentes descritos a continuación:

- Sistema Operativo (SO) Linux en cualquiera de sus variantes, entre estos las dos distribuciones más notables son Ubuntu y Debian por sus prestaciones, por el uso de comandos ShellScript en la herramienta.
- Servidor WEB Apache donde estarán ubicadas las aplicaciones y la capa de control del repositorio.
- El Servidor WEB debe ser capaz de compilar PHP en su versión 5 y a este debe estar incorporado el controlador o driver para manejar bases de datos en SQLite y MySQL.
- El sistema operativo debe contar con la herramienta aapt (*Android Application Package Tool*), herramienta que permite descompactar los paquetes de instalación o apks y de esta manera es posible obtener información relacionada a ellos.

Una vez definidos los requisitos, funcionales y no funcionales, es posible compenetrarse en el diseño del sistema, mediante diagramas de casos de uso del sistema, esclareciendo el camino de un método más interactivo gracias a las facilidades de comprensión que ofrecen los gráficos.

## **2.7 Diagramas de Casos de Uso del Sistema**

A continuación se ejemplifican mediante diagramas de casos de uso del sistema (CUS) expresado en la notación UML la solución propuesta. Estos diagramas exponen las acciones que los usuarios pueden realizar mediante la interacción con el sistema facilitando considerablemente la posterior implementación de estos.

En el diagrama expuesto en la FIG 2.3 se describen los casos de uso del Jaas (cliente) mientras que en el diagrama de la FIG 2.4 se expone las acciones relativas al Administrador a con el sistema.

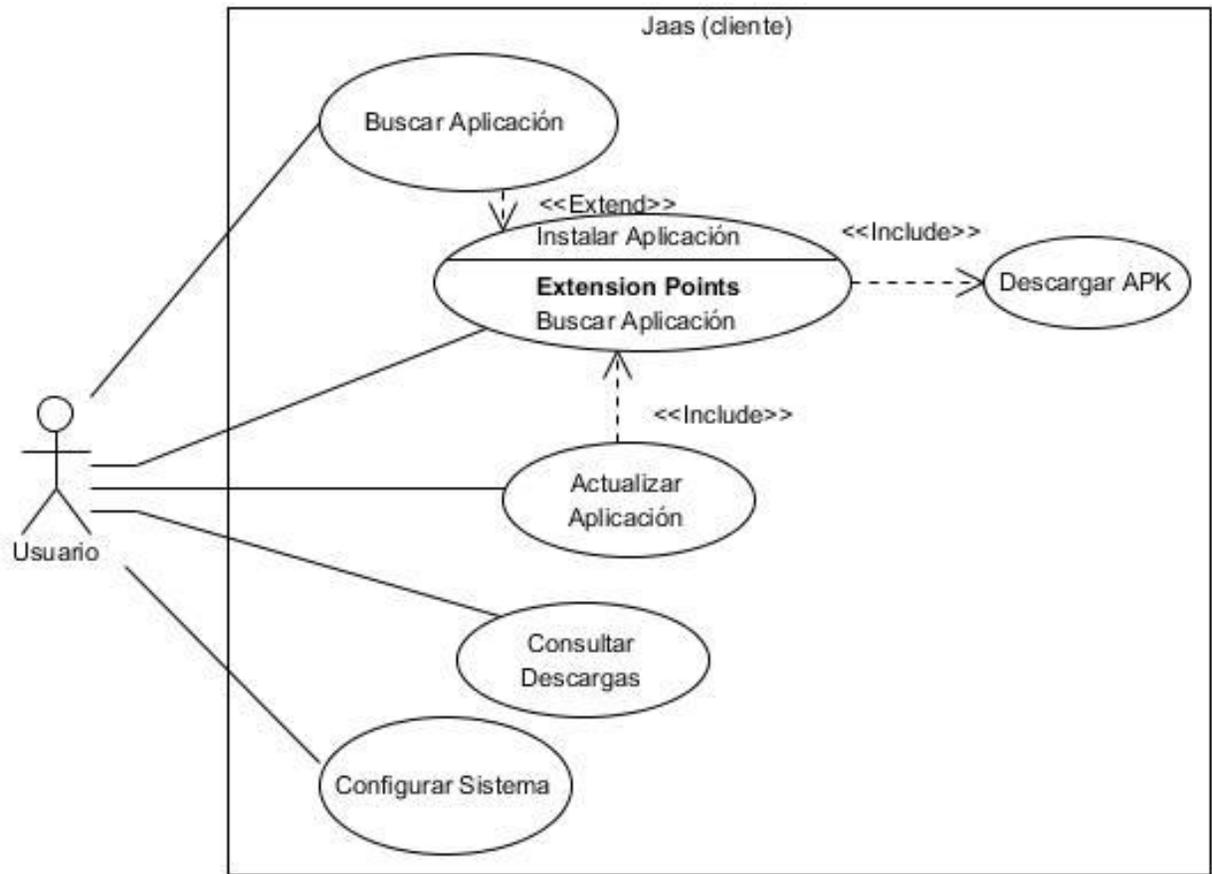


Figura 2.3 Diagrama de CU del Sistema Jaas (cliente).

En el expuesto diagrama de casos de uso del sistema de la FIG 2.3 se detallan las múltiples actividades que el usuario o cliente del Jaas puede acometer a con el sistema. A continuación se exponen las descripciones de cada uno de los casos de uso accionables por el usuario según destaca la metodología de desarrollo AUP plasmada en el capítulo primero.

Caso de Uso del Sistema Jaas(cliente) #1	
Nombre	Buscar Aplicación
Descripción	Como el nombre deja entrever permite al usuario navegar por el sistema buscando aplicaciones, este caso de uso va más allá de la simple búsqueda e incluye las funcionalidades relacionadas con el caso de uso como “Filtrar”.
Casos de uso asociados	Instalar Aplicación
Actores asociados	Usuario

**Tabla 2.4 Descripción del CUS Buscar Aplicación**

El caso de uso del sistema Buscar Aplicación, representado en la TABLA 2.4, interactúa con Instalar Aplicación, expuesto en la TABLA 2.5, debido que en muchas ocasiones se ejecuta tras realizar una acción relativa al CUS de búsqueda. Una vez se realiza la orden de instalar aplicación este, CUS #2 Instalar Aplicación, incluirá Descargar Aplicación.

Caso de Uso del Sistema Jaas(cliente) #2	
Nombre	Instalar Aplicaciones
Descripción	Es el principal caso de uso porque es en función el centro de todas y cada una de las tiendas de aplicaciones y aplicaciones similares. Este caso de uso del sistema es el encargado de ordenar la descarga de una aplicación determinada.
Casos de uso asociados	Buscar Aplicación, Actualizar Aplicación, Descargar APK

Actores asociados	Usuario
-------------------	---------

**Tabla 2.5 Descripción del CUS Instalar Aplicación**

Caso de Uso del Sistema Jaas(cliente)#3	
Nombre	Actualizar Aplicación
Descripción	Este incluye el Instalar Aplicación pues es una situación específica del instalar, aunque debe consultar con el dispositivo las aplicaciones instaladas.
Casos de uso asociados	Instalar Aplicación
Actores asociados	Usuario

**Tabla 2.6 Descripción del CUS Actualizar Aplicación**

En la TABLA 2.6 se describe formalmente el caso de uso del sistema Actualizar Aplicación, este es dependiente de las aplicaciones disponibles y las instaladas en el dispositivos. Luego, en la TABLA 2.7, se muestra la descripción del CUS Consultar Descargas y posteriormente Configurar Sistema, que mediante este caso de uso el usuario es capaz de configurar el sistema, como denota su nombre, según sus preferencias y permitiendo flexibilidad en la ubicación del servidor de aplicaciones.

Caso de Uso del Sistema Jaas(cliente) #4	
Nombre	Consultar Descargas
Descripción	Mediante este caso de uso el usuario puede acceder a las aplicaciones ya descargadas permitiendo si lo desea el reinstalar

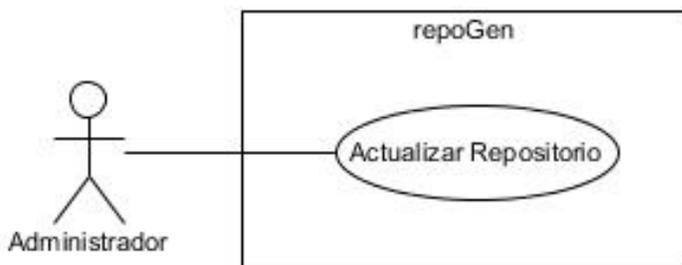
	una aplicación o eliminarla para optimizar el espacio libre en memoria.
Actores asociados	Usuario

**Tabla 2.7 Descripción del CUS**

Caso de Uso del Sistema Jaas(cliente) #5	
Nombre	Configurar Sistema
Descripción	El usuario puede configurar la dirección del repositorio, la forma inicial de ordenar las aplicaciones y limpiar los datos de la aplicación.
Actores asociados	Usuario

**Tabla 2.8 Descripción del CUS**

A continuación se exponen el diagrama de casos de uso del sistema y su descripción en la (FIG 2.4) y TABLA 2.9, Administrador, se describen a continuación en la tabla 2.8.



**Figura 2.4 Diagrama de CUS caso Administrador**

---

Caso de Uso del Sistema Administrador #1	
Nombre	Actualizar Repositorio
Descripción	El administrador es el encargado de actualizar las bases de datos de aplicaciones.
Actores asociados	Usuario

*Tabla 2.9 Descripción del CUS Administrador*

## **CAPÍTULO 3 - Herramientas, repositorio y su herramienta de manejo y aplicación cliente en Android.**

El siguiente capítulo muestra el proceso realizado en el desarrollo de las herramientas empleando las herramientas de diseño necesarias y tomando como fuente la modelación del negocio realizada en el Capítulo #2.

### **3.1 Arquitectura del Sistema**

El sistema está planteado con una arquitectura basada en el Modelo Vista Controlador (MVC) como la mayoría de los proyectos Android. Esto permite manejar de forma independientes las actividades encargadas de las vistas y las clases controladoras que como su nombre da a relucir están encargadas de controlar el funcionamiento interno del sistema. En el caso específico de Android el modelo vista-controlador tiene como principal bondad separar los datos de una aplicación, la interfaz de usuario y la lógica de negocios en tres componentes distintos que se relacionarán para al final tener como resultado nuestra aplicación.

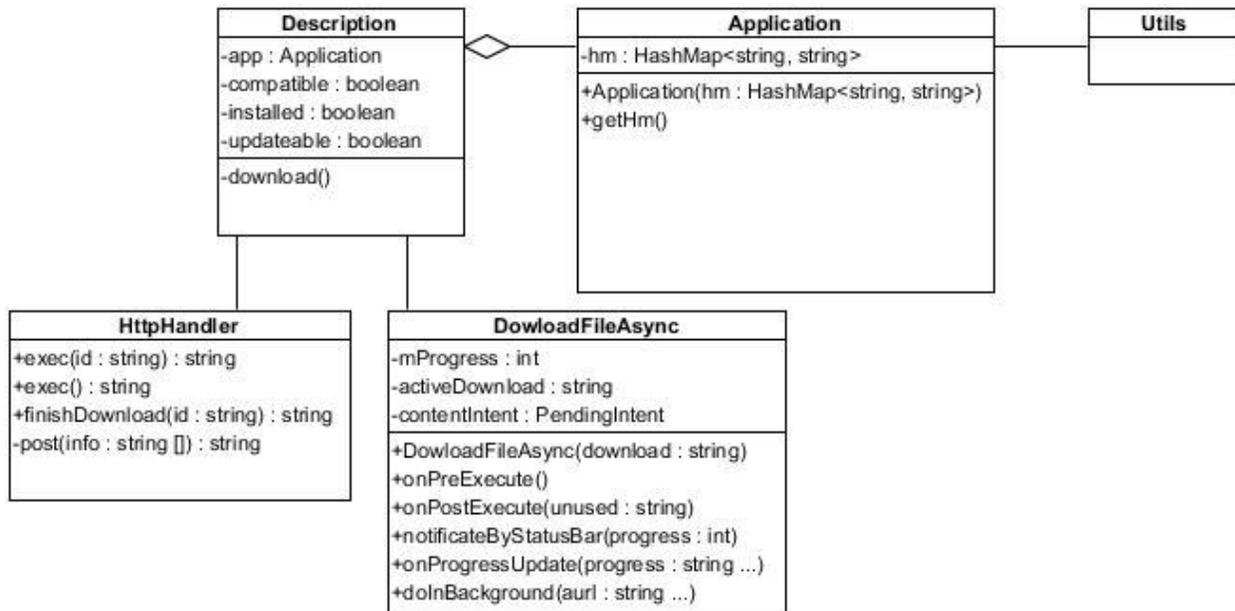
**Modelo:** Son las representaciones que se construyen basadas en la información con la que operará la aplicación. En Java, el modelo viene siendo análogo a los beans que tienen la particularidad de ser reutilizables y nos ayudan a cumplir con el proverbio de oro “*Don’t Repeat Yourself*” (DRY) haciendo las aplicaciones escalables. Esta parte del modelo también juega un papel importante en la decisión de qué pauta para almacenar información se utilice, el modelo elegido depende obviamente de las necesidades de información de la aplicación.

**Vista:** No es más que la interfaz con la que va a interactuar el usuario. En Android, las interfaces se construyen en XML. Construyendo un esqueleto en XML que equivale al HTML de un sitio. Posteriormente, con ayuda de estilos, que también escritos en XML, se puede empezar a dar formato de colores, posiciones, formato, etc. al esqueleto. Esto equivale a los CSS del lenguaje HTML.

**Controlador:** Finalmente nos topamos con el controlador que son todas esas clases que nos ayudarán a darle vida a esas interfaces bonitas que ya construimos y nos permitirán desplegar y consumir información de/para el usuario. Estos controladores se programan en lenguaje Java y son el núcleo de la aplicación.

### 3.2 Diagrama de clases de diseño del CUS Descargar APK

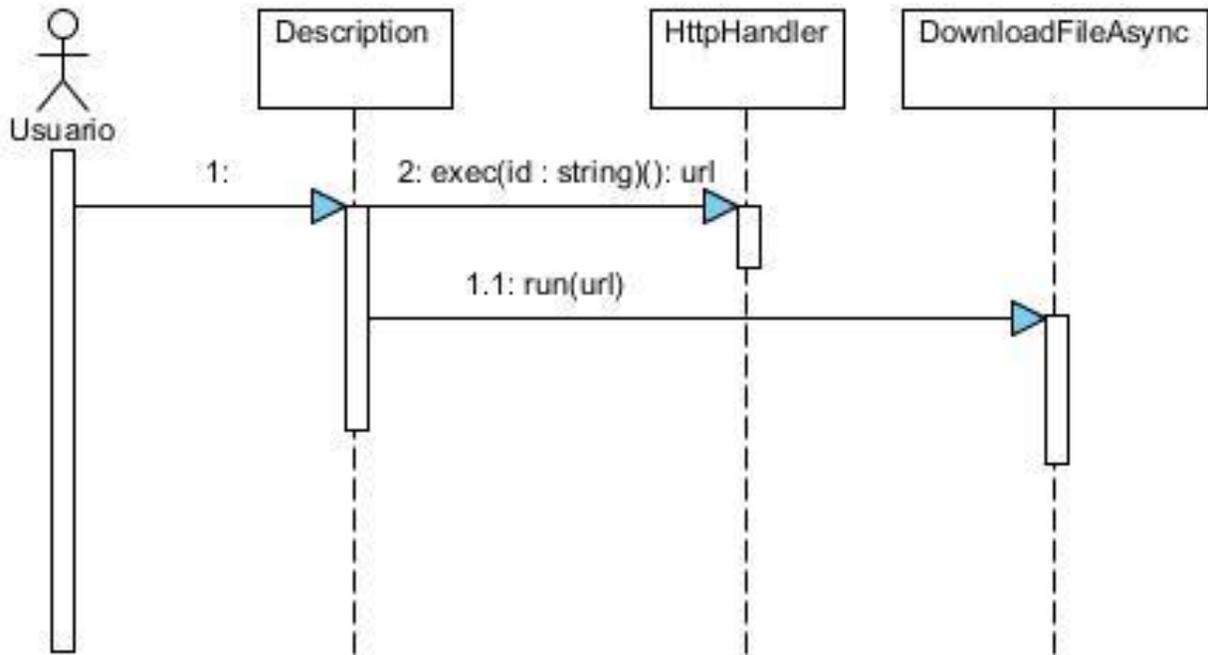
Se muestra seguidamente el diagrama de clases de diseño del sistema Jaas para el caso de uso “Descargar APK” en la Figura 3.1. El modelo parte de las necesidades de la clase Description, clase encargada de mostrar al usuario la información de la aplicación seleccionada, para conformar la descarga posteriormente realizada por la clase DownloadFileAsync. Cabe destacar que tanto la clase HttpHandler y Utils son clases puramente estáticas que permiten llamar a sus métodos sin necesidad de realizar instancias de las mismas, esta última solamente ofrece valores constantes para la conformación del HashMap comprendido en la clase Application por lo que se aprecia innecesario la inclusión de sus datos en el diseño.



*Fig 3.1 Diagrama de clases del diseño relativo al CUS Descargar APK*

### 3.3 Diagrama de secuencia del CUS Descargar APK

La aplicación consta con una serie de funcionalidades que toman forma a través de las pantallas y su interacción con las clases manejadoras de datos. En la Fig. 3.2 se muestran realizados por la aplicación para lograr ejecutar la funcionalidad que atiende la descarga de aplicaciones.



*Figura 3.2 Diagrama de Secuencia del CUS Descargar APK*

Una vez el usuario accede a la aplicación tiene a su alcance un listado de aplicaciones y seleccionando una incurre en el diagrama de secuencia que se expone en la Figura 3.2. Este inicia en la clase *Description*, encargada de mostrar al usuario información sobre la aplicación que está a punto de descargar, y posteriormente selecciona la opción de instalar, actualizar o reinstalar la aplicación. Este llamará a la clase *HttpHandler* para construir una dirección “url” para ordenarle a la clase *DownloadFileAsync* la descarga de la aplicación desde la ubicación determinada.

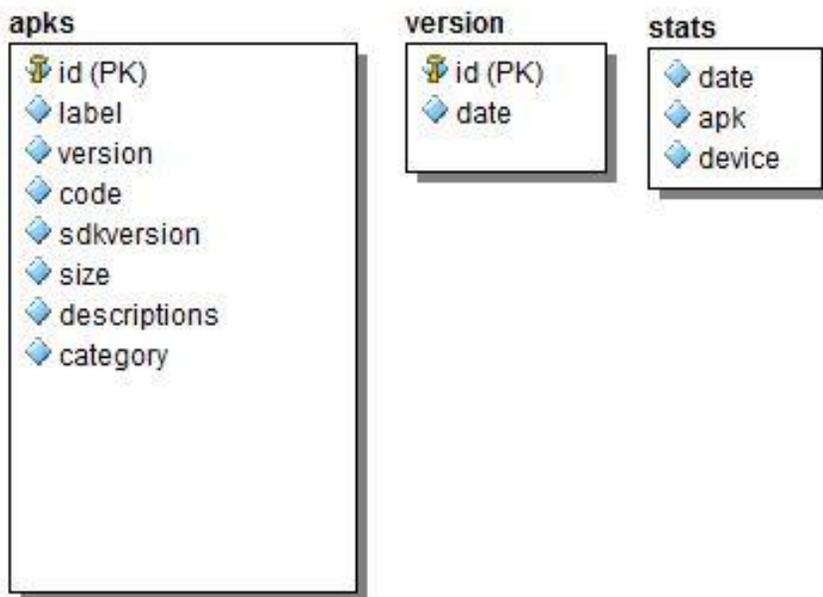
### **3.4 Tratamiento de errores**

Existen cuatro situaciones claves donde con mayor frecuencia se puede presenciar un error y todas ellas coinciden en el proceso de la descarga de una aplicación. El primer caso puede ocurrir por la falta de la aplicación en el repositorio ya sea por una mala manipulación del sistema archivos o por una entrada inválida en las bases de datos. El segundo caso por un error de descarga que puede suceder por “tiempo de espera agotado” o *timeout*, problemas preexistentes en el paquete de instalación o quizás fallos en la conexión que ocasionarían el llamado error de “tiempo de espera agotado”. Una tercera posibilidad puede ocurrir relacionada con la primera, dado al funcionamiento offline que permite la aplicación cliente si no se está conectado al servidor de aplicaciones o por algún motivo este no se está funcional estaríamos en presencia del mismo error y como cuarta situación está espacio insuficiente en memoria que está vinculado al segundo caso de error pues el proceso es interrumpido por problemas con la descarga del paquete de instalación.

Existen varias maneras para tratar con casos de errores como estos tres descritos anteriormente, básicamente se soluciona deteniendo el proceso encargado de realizar la descarga del archivo y notificando al usuario. En el caso realizado se detiene automáticamente el proceso en segundo plano que se encontraba realizando la descarga y el usuario es notificado de ello mediante una notificación del sistema denominada Toast.

### **3.5 Diseño de la base de datos**

Para el diseño de las bases de datos (BD) es muy importante tener en cuenta el rol que deben tomar estas en los distintos sistemas. En esta ocasión solamente se necesitaba almacenar información de una manera modelo permitiendo múltiples transferencias y acceso a los datos de manera ágil y eficiente. Por ello se decide dividir el diseño en dos modelos físicos, uno para una base de datos en MySQL, denominada “repo” a partir de este punto, habilitada para el control de aplicaciones en el servidor y estadísticas de descargas y otro para una base de datos en SQLite que el cliente descarga con el listado de las aplicaciones existentes en el repositorio. En la Figura 3.3 se puede observar modelo físico empleado en la construcción de la base de datos repo.



*Figura 3.3 Modelo físico de la base de datos “repo”*

Como se puede presenciar en el modelo representado en la Figura 3.3 no existe interrelación entre las tablas, pues el objetivo es únicamente almacenar información de manera flexible y eficiente de las aplicaciones.

Seguidamente en la Figura 3.4 se muestra el modelo físico de la base de datos diseñada para SQLite, denominada “index” desde este entonces, creada para ser descargada por los usuarios de la herramienta Jaas evitando así largos tiempos de espera que causa el poblar una base de datos mediante consulta desde un dispositivo móvil, debido a que estos no poseen la capacidad de procesamiento de un ordenador común.



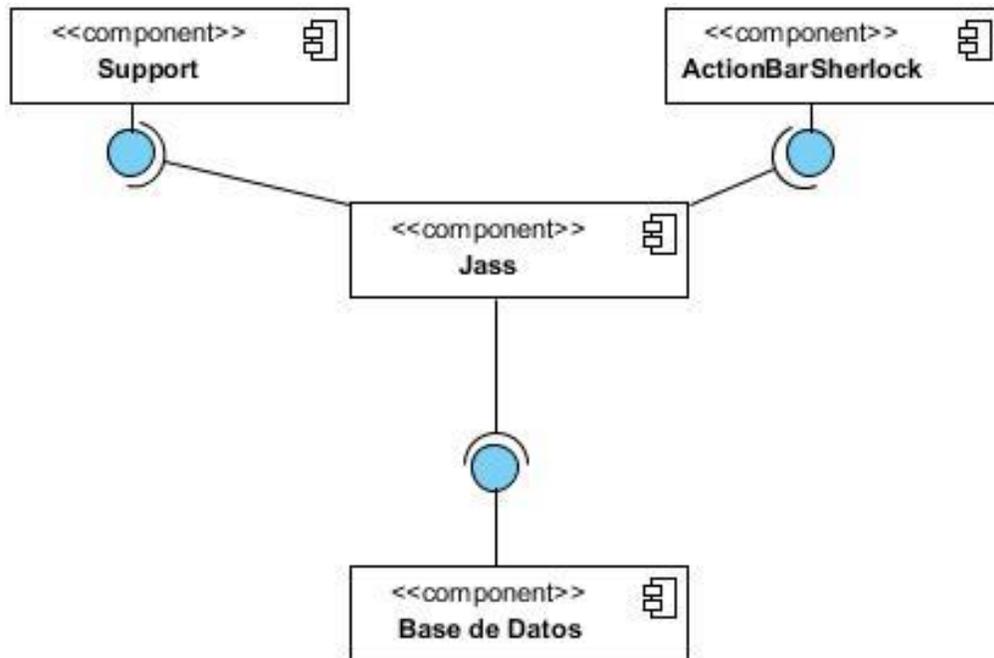
*Figura 3.4 Modelo físico de la base de datos “index”*

El modelo físico ejemplificado muestra el diseño que poseerá la base de datos de SQLite, que no es más que una copia de la tabla “apks” de la BD “repo” y como SQLite no cuenta con un gestor, es solo un archivo encriptado, de fácil acceso, permite que el cliente de la aplicación Jaas acceder a las listas de aplicaciones con facilidad.

### **3.6 Modelo de componentes**

La Fig. 3.7 muestra el campo de acción que posee el usuario final del sistema. La aplicación cliente asume el juicio final, tomando en cuenta que es de libre distribución, dicho esto no quiere decir que el resto del sistema no lo sea, solo que existe una política más restringida con respecto a los datos que manejan. El APK, no es la aplicación final corriendo en el celular del usuario, es el paquete que contiene toda la información necesaria para su instalación. Estos componentes son los futuros mediadores de la aplicación, donde interactúan mediante interfaces para echar a andar

la aplicación la biblioteca *Support* y para lograr la compatibilidad entre versiones del OS Android, la *Sherlock*. También está la base de datos en el servidor con la que interactúa mediante la clase *HttpHandler* que funciona como interfaz de comunicación. Esta está descrita anteriormente en el diagrama de clases del diseño y en el diagrama de secuencia, y su función en el presente diagrama de componentes es controlar la intercomunicación entre las bases de datos del servidor y la aplicación cliente *Jaas*.



*Figura 3.5 Diagrama de componentes de la aplicación cliente Jaas*

### **3.7 Diagrama de despliegue**

El despliegue, como se muestra en la Fig. 3.6, consta de con una distribución de sus activos, proponiendo un manejo clásico de sistemas basados en usuarios. Las aplicaciones Android están empotradas en los Smartphone que acceden a sus respectivas bases de datos, mientras hacen peticiones a un servidor para gestionar la descarga de aplicaciones.



*Figura 3.6 Diagrama de despliegue de componentes del sistema*

## **CAPÍTULO 4 - Planificación del sistema.**

El siguiente capítulo muestra el proceso para el cálculo de costo y tiempo basado en los casos de uso para las funcionalidades desarrolladas.

### **4.1. Planificación basada en el método de estimación por Caso de Uso**

La planificación de este sistema está basada en el tiempo de duración del proyecto, los recursos consistentes y en el personal libre de costo monetario a utilizar. Además se tiene en cuenta los horarios de trabajo con mayor rendimiento, y las estrategias a seguir para los cortes de entregas.

### **4.2. Estimación basada en Casos de Uso.**

La estimación basada en Casos de Uso es un método desarrollado por Gustav Karner de Objectory AB, teniendo su posterior consolidación en otros autores. Para aplicar este método, hay que asignar pesos a los factores que afecten el desarrollo del proyecto, y de esta manera, contabilizar el tiempo total estimado.

#### **Estimación del Esfuerzo Basada en Casos de Uso.**

##### **Cálculo de Puntos de Casos de Uso sin Ajustar (UUCP).**

Este valor se calcula a partir de la siguiente ecuación:

$$UUCP = UAW + UUCW \quad \text{donde,}$$

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Determinación del factor de peso de los actores sin ajustar (UAW).

Este valor se calcula mediante el análisis de la cantidad de Actores y la complejidad que presenta para ellos el sistema. La complejidad de los actores se toma, partiendo del contexto, sí se trata de una persona o de otro sistema, y además se evalúa la forma en que interactúan, en la Tabla 4.1, aparece descrito escaso de este sistema.

Tipo de actor	Descripción	Factor de peso	Número de actores	Resultado
Simple	Interacción entre sistemas mediante una interfaz de programación.	1	0	0
Promedio	Interacción entre sistemas mediante un protocolo o una interfaz basada en texto.	2	1	2
Complejo	Interacción entre una persona y el sistema mediante una interfaz gráfica.	3	1	3
			<i>Total</i>	5

*Tabla 4.1 Factores de peso de los actores*

Teniendo como resultado, que  $UAW = 5$

**Determinación del factor de peso en los casos de uso sin ajustar (UUCW).**

Este valor se calcula partiendo de la cantidad y complejidad de los Casos de Uso del sistema. La complejidad se establece por la cantidad de transacciones efectuadas, esto se describe como una secuencia de actividades atómicas, donde se ejecutan todas o desaparecen sí una de ellas rompe su enlace. Tabla 4.2

Tipo de caso de uso	Descripción	Factor de peso	Número de Casos de Uso	Resultado
Simple	1-3 Transacciones	5	5	25
Promedio	4-7 Transacciones	10	0	0
Complejo	Mayor de 8 Transacciones.	15	0	0

	<i>Total</i>	25
--	--------------	----

**Tabla 4.2 Factores de peso de los casos de uso**

$$UUCW = 25$$

Por tanto:

$$UUCP = UAW + UUCW$$

$$UUCP = 5 + 25$$

$$UUCP = 30$$

**Cálculo de Puntos de Casos de Uso ajustados.**

Para calcular los Puntos de Casos de Uso ajustados, toma la siguiente ecuación:

$$UCP = UUCP \times TCF \times EF \text{ donde,}$$

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

**Determinación del factor de complejidad técnica (TCF).**

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinantes en la complejidad técnica del sistema. Los factores se cuantifican con un valor entre 0 y 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. Tabla 4.3

Número de factor	Descripción	Peso	Valor	Factor	Comentario
T1	Sistema Distribuido	2	2	4	<i>El sistema consta con una aplicación cliente y otra administración.</i>
T2	Tiempo de respuesta	1	2	2	<i>Se trabaja con información que acapara, relativamente poco espacio.</i>

T3	Eficiencia por el usuario	1	3	3	<i>Se puede mejorar la administración.</i>
T4	Proceso interno complejo	1	3	3	<i>El sistema solo gestiona datos.</i>
T5	Reusabilidad	1	3	3	<i>Parcial, algunas bibliotecas son reutilizables.</i>
T6	Facilidad de instalación	1	3	3	<i>Muy sencilla, solamente es necesaria su ejecución en un dispositivo apropiado.</i>
T7	Facilidad de uso	1	5	5	<i>Para el usuario común si, para el administrador, debe auxiliarse de la ayuda al principio, aun así no es tan complejo.</i>
T8	Portabilidad	2	5	10	<i>Las APK tienen poco peso de almacenamiento, pueden ser descargadas muy rápido.</i>
T9	Facilidad de cambio	1	5	5	<i>El sistema está presto a cambios de todo tipo por su flexibilidad.</i>
T10	Concurrencia	1	5	5	<i>Múltiples usuarios pueden concurrir descargando aplicaciones</i>
T11	Objetivos especiales de seguridad	1	5	5	<i>El administrador del sistema es el único que tendrá acceso a cambiar los datos relevantes.</i>
T12	Acceso directo a terceras partes	1	2	2	<i>Ninguno, es una aplicación independiente.</i>

T13	Facilidades especiales de entrenamiento a usuarios finales	1	1	1	<i>Todo se puede obtener mayor ayuda contactando a los responsables, descritos en la interfaz "About".</i>
			<i>Total Factor</i>	<i>51</i>	

**Tabla 4.3. Factores de complejidad técnica**

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 * \sum(\text{Peso}_i \times \text{Valor asignado}_i)$$

$$TCF = 0.6 + 0.01 * 51$$

$$TCF = 1.11$$

#### **Determinación del factor ambiente (EF).**

La determinación del factor ambiente involucra las habilidades y el entrenamiento del grupo involucrado en el desarrollo del proyecto. Tabla 4.4

Número del factor	Descripción	Peso	Valor	Factor	Comentario
E1	Familiaridad con el modelo del proyecto usado.	1	4	4	<i>La concepción del modelado del proyecto puede ser modificada para mejoras en próximas versiones.</i>
E2	Experiencia en la aplicación	1	4	4	<i>Poca en el desarrollo de aplicaciones Android.</i>
E3	Experiencia OO.	1	3	3	<i>Media.</i>
E4	Capacidad del analista líder.	0.5	1	0.5	<i>No existe analista líder.</i>

E5	Motivación.	1	5	5	<i>Existe gran motivación para el desarrollo del proyecto.</i>
E6	Estabilidad de los requerimientos.	2	5	10	<i>El sistema se encuentra sujeto a mejoras pero es totalmente estable.</i>
E7	Personal media jornada.	-1	0	0	<i>Se trabajará a tiempo completo.</i>
E8	Dificultad en lenguaje de programación.	-1	3	-3	Lenguaje java.
			<i>Total</i>	<i>23.5</i>	

**Tabla 4.4 Factores de ambiente**

El factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$EF = 1.4 - 0.03 * 23.5$$

$$EF = 2.105$$

Cálculo de los Puntos de de Casos de Uso Ajustados:

$$UCP = UUCP * TCF * EF$$

$$UCP = 57 * 1.11 * 2.105$$

$$UCP = 133.18335$$

**Cálculo del esfuerzo.**

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF \text{ donde:}$$

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de casos de uso ajustados.

CF: Factor de conversión (19,2 horas-hombre).

$$E = 133.18335 * 20$$

$E \approx 2560$

En la Tabla 4.5 se muestra la relación entre las actividades realizadas y el porcentaje que representa en esfuerzo.

Tabla 4.5. Distribución genérica del esfuerzo.

Actividad	Porcentaje
Análisis	20.00%
Diseño	10.00%
Programación	50.00%
Pruebas	10.00%
Sobrecarga(otras actividades)	10.00%

**Tabla 4.5 Distribución genérica del esfuerzo**

La Tabla 4.6 muestra la relación entre las actividades y el tiempo aproximado de esfuerzo.

Actividad	Horas
Análisis	512
Diseño	256
Programación	1280
Pruebas	256
Sobrecarga(otras actividades)	256
<i>Total</i>	2560

**Tabla 4.6 Distribución real del esfuerzo**

Cálculo del esfuerzo total:

$$E_{\text{Total}} = 2560 \text{ horas /hombre}$$

Cálculo del tiempo de desarrollo:

$$T_{\text{Desarrollo}} = E_{\text{Total}} / \text{CHTotal} \quad \text{CHTotal: Cantidad de hombres} = 1$$

$$T_{\text{Desarrollo}} = 2560 \text{ horas}$$

Considerando que se trabajan 8 horas diarias:

$T_{\text{Desarrollo}} = T_{\text{Desarrollo}} / 8 \text{ horas/día}$

$T_{\text{Desarrollo}} = 2560 \text{ horas} / 8 \text{ horas/día}$

$T_{\text{Desarrollo}} = 320 \text{ días aproximadamente}$

Cálculo del costo:

$\text{CostoTotal} = E_{\text{Total}} * 2 * TH$       TH: Tarifa horaria (= 1.031)

$\text{CostoTotal} = 2560 * 1 * 1.031$

$\text{CostoTotal} = 2639.36$

#### **4.2.1. Análisis de Costo – Beneficio.**

El sistema como el propuesto ofrece los siguientes beneficios:

Beneficios tangibles:

Son los que se miden por el alcance que tengan en el impacto del ahorro de recursos materiales.

El sistema no actúa directamente en este ámbito, pero si se puede referir a ello, en los casos en los donde se anotan direcciones, números telefónicos o incluso pequeños mapas de orientación.

Beneficios intangibles:

El campo de acción del sistema está dirigido a este tipo de beneficios no materiales, algunos de ellos se muestran a continuación:

- Los usuarios conocerán con mayor detalle la UCLV.
- Forma parte de la toma de decisiones de los usuarios, ayudándolos con los trámites o búsqueda de información.
- Mejora la calidad y rapidez del proceso de búsqueda en cuanto a tiempo, procesamiento y organización de la información.
- Ayuda al visitante o residente, con una guía universitaria de fácil navegación.
- Ayuda en la divulgación de las instalaciones universitarias en ámbitos externos a la misma.

El sistema se desarrolla sin costos materiales, utilizando los medios que ofrece la universidad así como la ayuda de dicha comunidad. El capital humano utilizado, se concentró en la ayuda de la

conformación de modelos usados para la comprensión del funcionamiento universitario y de esta forma brindar una solución factible.

## Conclusiones

En cuatro capítulos de este trabajo se superó, explicadas las bases que sustentan el dominio teórico, el problema que supone la no existencia de un mecanismo automatizado que permita a los usuarios poseedores de dispositivos móviles con tecnología Android descargar e instalar aplicaciones. La implementación de un sistema compuesto por un repositorio de aplicaciones Android, esto es logrado con una aplicación para administrar dicho repositorio y una aplicación cliente Android que permita buscar e instalar las aplicaciones existentes en dicho repositorio, esto constituye una alternativa viable a la utilización de la tienda virtual Google Play Store. De acuerdo a las aplicaciones desarrolladas como parte del sistema, se concluye:

- La creación de una repositorio de aplicaciones que permite ser utilizado por una aplicación Android de forma offline tiene un peso fundamental en el resultado final obtenido a partir de la implementación del sistema propuesto.
- La herramienta de administración del repositorio cumple con sus funciones de mantener actualizada la información y las aplicaciones existentes en este último.
- La aplicación Android *Jaas* cumple con su rol de consumir la información y las aplicaciones brindadas por el repositorio.
- La aplicación Android *Jaas* se encuentra lista para su despliegue final y libre utilización.

## **Recomendaciones**

- Continuar trabajando en las diversas capas de la aplicación para obtener mejores resultados y versiones más eficientes.
- Realizar una herramienta capaz de generar reportes con la información adicional que se gestiona conjuntamente con la descarga de aplicaciones.

## Referencias

- Google Play Store* [Online]. Google Inc. Available: <http://play.google.com/store> [Accessed Diciembre 12 2013].
- Official Apple Store* [Online]. Apple Inc. Available: <http://store.apple.com> [Accessed Enero 04 2014]. 2013.
- Rup Agil Ingenieria de Software* [Online]. Available: [http://ingenieriadesoftware.mex.tl/52788\\_Rup-Agil.html](http://ingenieriadesoftware.mex.tl/52788_Rup-Agil.html) [Accessed Marzo 20 2014].
- JAMES GOSLING, B. J., GUY STEELE Y GILAD BRACHA 2005. *The Java language specification*, Addison-Wesley.
- VALDEOLMILLOS, C. 2014. *El smartphone, el dispositivo más utilizado para acceder a la Red La AIMC presenta los resultados de la 16 Encuesta a Usuarios de Internet en España* [Online]. Available: <http://www.publico.es/504889/el-movil-desbanca-al-resto-de-equipos-en-la-conexion-a-internet> [Accessed Marzo 17 2014].
- VALERO, C. 2013. *Las mejores tiendas de aplicaciones legítimas para Android* [Online]. Available: <http://www.adslzone.net/article11496-las-mejores-tiendas-de-aplicaciones-legitimas-para-android.html> [Accessed Diciembre 18 2013].
- WIKIPEDIA. 2014. *Extensible Markup Language* [Online]. Available: [http://es.wikipedia.org/wiki/Extensible\\_Markup\\_Language](http://es.wikipedia.org/wiki/Extensible_Markup_Language) [Accessed Marzo 19 2014].