

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA  
Y TELECOMUNICACIONES**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL  
TÍTULO DE INGENIERÍA**

**DESARROLLO E IMPLEMENTACIÓN DE SOFTWARE  
DE RECONOCIMIENTO DE DÓLARES AMERICANOS  
DIRIGIDO A PERSONAS CON DISCAPACIDAD VISUAL  
UTILIZANDO TELÉFONOS MÓVILES INTELIGENTES  
CON SISTEMA OPERATIVO ANDROID**

**CHRISTIAN ROBERTO CATACTA LLIVE  
CARLOS ANDRÉS GUAITA AYALA**

**Sangolquí - Ecuador**

**2011**

## **CERTIFICACIÓN**

Certificamos que el presente proyecto de grado fue realizado en su totalidad por los Sres. Catacta Llive Christian Roberto y Guaita Ayala Carlos Andrés bajo nuestra dirección.

---

Ing. Fabián Sáenz

**DIRECTOR**

---

Ing. Paúl Bernal

**CODIRECTOR**

## **RESUMEN**

El presente documento trata sobre el desarrollo de un software de Reconocimiento de la Denominación de Dólares Americanos dirigido a personas con discapacidad visual, el cual es diseñado para Teléfonos Móviles Inteligentes que funcionen bajo el Sistema Operativo Android de Google.

Para la elaboración de dicha aplicación previamente se hizo un estudio de la penetración de los Teléfonos Móviles Inteligentes y sus Sistemas Operativos en el Ecuador. Posteriormente se profundizó en la investigación del Sistema Operativo Android, el cual es basado sobre el Kernel de Linux, que permite desarrollar aplicaciones sin costos de licencias por ser software libre.

La aplicación tiene dos etapas, la primera es la de aprendizaje, la cual obtiene un banco de imágenes de los rostros que viene en cada uno de los billetes al lado frontal, toda esta etapa es realizada en Matlab; la segunda etapa es la de reconocimiento, desarrollada en Eclipse IDE bajo el lenguaje de programación Java y utilizando la API de Android, para el reconocimiento la aplicación utiliza un algoritmo principal de Análisis de Componentes Principales y filtros basado en Procesamiento Digital de Imágenes, obteniendo un software prototipo, base para futuros proyectos en el Sistema Operativo Android.

## **DEDICATORIA**

Este trabajo de esfuerzo, dedicación y constancia, va dedicado a todas esas personas que han luchado en la vida con cierta desventaja, y han salido adelante con gran entusiasmo y esmero; es por ello y muchas circunstancias más, que este proyecto va dedicado a las personas con Discapacidad Visual de nuestro país y a todas aquellas personas que extienden la mano a los que más lo necesitan.

## **AGRADECIMIENTOS**

Un agradecimiento muy especial, a todas esas personas que de una u otra manera contribuyeron a la realización de este trabajo de investigación.

En especial agradezco a Dios, que es la Fe y la Fuerza Interna que me permite realizar todas las actividades por el camino del bien, en segundo lugar agradezco infinitamente a mis Padres que siempre me brindan su amor incondicional, su apoyo para seguir adelante y su ejemplo de trabajo, esfuerzo y perseverancia en todo momento de la vida, y en tercer lugar a mi hermano, novia y amigos que hacen de mi vida un lugar agradable con su presencia, consejos y grandes momentos.

Christian Catacta LLive.

## **AGRADECIMIENTOS**

Agradezco a mi mamá Piedad por todo el apoyo brindado, por enseñarme que el verdadero sentido de la vida se encuentra en hacer feliz y apoyar a todos nuestros seres queridos y que el mayor tesoro que un hombre puede tener es estar rodeado por personas que lo aprecian y lo aman.

A mis tíos que supieron ser una guía en el momento adecuado y que estuvieron presentes cuando más lo necesitaba.

A mi madre Mónica que a la distancia siempre me apoyo y se encontró formas para ser una guía en mi camino.

Y a mis amigos por estar conmigo en las buenas y en las malas, por compartir todos estos años de estudio que además de los conocimientos nos dejaron experiencias de vida.

Andrés Guaita Ayala.

## PRÓLOGO

El Proyecto consiste en hacer una aplicación que funcione en teléfonos móviles inteligentes que posean el sistema operativo Android de Google, esta aplicación permite reconocer la denominación de Dólares Americanos mediante la cámara del dispositivo móvil y presentarlo en modo de audio para las personas con discapacidad visual.

La investigación de este proyecto se ha desarrollado en base al proyecto realizado en el año 2010, que consistía en un prototipo de reconocimiento de Dólares Americanos el cual fue desarrollado como tesis de grado por estudiantes de la Escuela Politécnica del Ejército. Aquel proyecto dejó varios puntos pendientes y la posibilidad de ser mejorado, es por eso que este proyecto tiene como principal objetivo el de migrar la aplicación hecha para dispositivos móviles Nokia con sistema operativo Symbian hacia un sistema operativo de nueva generación y de código abierto que funcione en teléfonos móviles inteligentes llamado Android.

Para la elaboración de este proyecto de grado se realizó un trabajo de investigación que se conforman de siete capítulos detallados de la siguiente manera:

El capítulo 1, indica la información obtenida de la investigación sobre las personas con discapacidad visual, mostrando sus estadísticas, organismos y leyes que rigen actualmente en el Ecuador, además el rol que cumple la tecnología en especial la tecnología celular con dispositivos en beneficio de las personas no videntes; al final de este capítulo se enfoca el problema del reconocimiento de la denominación de Dólares Americanos por el cual se debe esta investigación, proponiendo el alcance y propósito de la aplicación a realizarse.

El capítulo 2, se detalla el tema de los teléfonos móviles inteligentes desde sus inicios hasta la actualidad, la evolución de sus características tanto en hardware como en software, el mercado de los teléfonos móviles inteligentes en América Latina y por último, los sistemas operativos que utilizan estos dispositivos con una breve comparación de sus características.

En el capítulo 3, se muestra la investigación sobre el nuevo sistema operativo para teléfonos móviles inteligentes llamado Android de la compañía Google, indicando sus inicios en el mundo de la tecnología celular, su arquitectura y sus principales características de sus diversas versiones y distribuciones de su plataforma.

El capítulo 4, explicará los temas más relevantes usados para la programación de la aplicación de reconocimiento utilizando el SDK de Android proporcionado por Google y herramientas como el IDE Eclipse y el lenguaje de programación Java para crear aplicaciones sencillas que puedan introducir al lector en el mundo del desarrollo de aplicaciones para Android.

En el capítulo 5, se detallan los conceptos y procedimientos utilizados para el diseño del software de reconocimiento de la denominación de Dólares Americanos. Dichos conceptos abarcan teoría de Reconocimiento Patrones, herramientas de Procesamiento Digital de Imágenes y Análisis de Componentes principales, siendo estos temas la base fundamental sobre la cual se desarrollará el software. En la parte final se expondrá la forma en la cual todos estos conceptos se conjugan para obtener el diseño del software.

En el capítulo 6, se describe el desarrollo e implementación del software de reconocimiento, para lo cual se considera los conceptos de diseño del capítulo 5 los cuales son puestos en desarrollo, además se explica la implementación del software en el teléfono móvil inteligente y se hace un análisis de desempeño y resultados para corregir, mejorar y optimizar el software para el uso de personas no videntes.

Finalmente en el capítulo 7, se enumeran las conclusiones obtenidas de todo el trabajo de investigación, además se mencionan las recomendaciones más relevantes del proyecto permitiendo dejar planteado un tema abierto para más investigación referente al sistema operativo Android. Se muestra además los anexos obtenidos de los diferentes capítulos y todo el código de programación de los distintos lenguajes para el desarrollo de la aplicación de reconocimiento.

## ÍNDICE DE CONTENIDOS

<b>ÍNDICE DE CONTENIDOS</b> .....	<b>1</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>1</b>
<b>INDICE DE FIGURAS</b> .....	<b>1</b>
<b>GLOSARIO</b> .....	<b>1</b>
<b>CAPITULO I</b> .....	<b>1</b>
<b>PERSONAS CON DISCAPACIDAD VISUAL</b> .....	<b>1</b>
<b>1.1 INTRODUCCIÓN</b> .....	<b>1</b>
<b>1.1.1 Población no vidente en el Ecuador</b> .....	<b>1</b>
<b>1.1.2 Estadísticas del CONADIS</b> .....	<b>2</b>
<b>1.2 EL ROL DE LA TECNOLOGÍA</b> .....	<b>3</b>
<b>1.2.1 Tecnología Celular en beneficio de las personas no videntes</b> .....	<b>4</b>
<b>1.3 IDENTIFICACIÓN DE OBJETOS DE PERSONAS NO VIDENTES</b> .....	<b>5</b>
<b>1.3.1 Identificación de la denominación de billetes</b> .....	<b>5</b>
<b>1.3.2 Aplicación celular de identificación de billetes basada en Android SO</b> .....	<b>6</b>
<b>1.4 ALCANCE DEL PROYECTO</b> .....	<b>6</b>
<b>CAPITULO II</b> .....	<b>8</b>
<b>TELÉFONOS MÓVILES INTELIGENTES</b> .....	<b>8</b>
<b>2.1 GENERALIDADES</b> .....	<b>8</b>
<b>2.2 TELÉFONOS MÓVILES INTELIGENTES</b> .....	<b>9</b>
<b>2.2.1 Evolución Histórica</b> .....	<b>10</b>
<b>2.3 CARACTERÍSTICAS DE LOS TELÉFONOS MÓVILES INTELIGENTES</b> .....	<b>11</b>
<b>2.3.1 Memoria interna</b> .....	<b>12</b>
<b>2.3.2 Tarjeta SIM</b> .....	<b>12</b>
<b>2.3.3 Evidencia Digital</b> .....	<b>13</b>
<b>2.3.4 Requisitos</b> .....	<b>14</b>
<b>2.4 MERCADO DE TELÉFONOS MÓVILES INTELIGENTES</b> .....	<b>16</b>
<b>2.5 SISTEMAS OPERATIVOS DE TELÉFONOS MÓVILES INTELIGENTES</b> .....	<b>18</b>

2.5.1	Android OS.....	19
2.5.2	Apple iOS .....	19
2.5.3	BlackBerry OS .....	20
2.5.4	Symbian OS.....	20
2.5.5	Windows Mobile OS.....	21
2.6	COMPARACIÓN ENTRE SISTEMAS OPERATIVOS.....	21
2.6.1	Detalles básicos .....	21
2.6.2	La interfaz de usuario .....	22
2.6.3	Funcionamiento .....	23
2.6.4	Desarrollo de terceros.....	24
<b>CAPITULO III .....</b>		<b>26</b>
<b>SISTEMA OPERATIVO ANDROID .....</b>		<b>26</b>
3.1	INTRODUCCIÓN.....	26
3.2	QUÉ ES ANDROID .....	28
3.3	ARQUITECTURA .....	28
3.3.1	Applications.....	29
3.3.2	Application Framework.....	30
3.3.3	Libraries .....	31
3.3.4	Android Runtime .....	32
3.3.5	Linux Kernel .....	33
3.4	CARACTERÍSTICAS IMPORTANTES INTEGRADAS EN ANDROID.....	34
3.5	VERSIONES DE LA PLATAFORMA ANDROID .....	34
3.5.1	Android 1.5 (Cupcake).....	35
3.5.2	Android 1.6 (Donut).....	36
3.5.3	Android 2.0 / 2.1 (Eclair).....	37
3.5.4	Android 2.2 (Froyo) .....	38
3.5.5	Android 2.3 (Gingerbread) .....	40
3.5.6	Android 3.x.....	41
3.6	DISTRIBUCIONES DE LAS VERSIONES DE ANDROID.....	42
3.6.1	Distribución actual.....	42
3.6.2	Android API Level .....	43
<b>CAPITULO IV .....</b>		<b>45</b>
<b>PROGRAMACIÓN EN ANDROID.....</b>		<b>45</b>
4.1	INTRODUCCIÓN.....	45

<b>4.2 PREPARACIÓN DEL ENTORNO DE DESARROLLO PARA ANDROID.....</b>	<b>47</b>
4.2.1 Requerimientos previos.....	47
4.2.2 Eclipse IDE .....	47
4.2.3 SDK de Android .....	48
4.2.4 El ADT .....	49
<b>4.3 INTRODUCCIÓN AL DESARROLLO PARA ANDROID .....</b>	<b>52</b>
4.3.1 Tipos de aplicaciones de Android.....	52
4.3.2 Componentes de una aplicación Android.....	54
4.3.3 Interfaces de usuario para Android.....	55
4.3.4 El archive AndroidManifest.xml .....	58
4.3.5 Recursos de la Aplicación .....	61
4.3.6 Ciclo de vida de una Aplicación Android.....	67
4.3.7 Herramientas para el desarrollo en Android.....	70
<b>4.4 DESARROLLO DE APLICACIONES ANDROID.....</b>	<b>76</b>
4.4.1 Creación de un nuevo proyecto de Android.....	76
4.4.2 Estructura de un proyecto Android.....	81
4.4.3 Creación de un Activity.....	84
4.4.4 Creación de una Interfaz de Usuario .....	85
4.4.6 Trabajando en segundo plano .....	94
4.4.7 Utilización de bases de datos SQLite y Content Providers .....	96
4.4.8 Acceso al Hardware mediante Android.....	97
<b>CAPITULO V.....</b>	<b>107</b>
<b>DISEÑO DEL SOFTWARE DE RECONOCIMIENTO DE LA</b>	
<b>DENOMINACIÓN DE DÓLARES AMERICANOS.....</b>	<b>107</b>
<b>5.1 INTRODUCCIÓN.....</b>	<b>107</b>
<b>5.2 RECONOCIMIENTO DE PATRONES.....</b>	<b>107</b>
5.2.1 Concepto de Reconocimiento de Patrones.....	107
5.2.2 Enfoques del Reconocimiento de Patrones .....	108
5.2.3 Sistema Básico de Reconocimiento.....	109
<b>5.3 PROCESAMIENTO DIGITAL DE IMÁGENES.....</b>	<b>111</b>
5.3.1 Imagen Digital.....	111
5.3.2 Modelos de Color .....	112
5.3.3 Histograma de una Imagen.....	117
5.3.4 Submuestreo de imágenes .....	121

5.3.5	Técnicas de Filtrado .....	124
5.4	<b>ANÁLISIS DE COMPONENTES PRINCIPALES .....</b>	<b>127</b>
5.4.1	Calculo de Eigenfaces .....	128
5.4.2	Proyección y reconstrucción de una imagen utilizando eigenfaces .....	130
5.4.3	Reconocer una cara.....	131
5.5	<b>DISEÑO DEL SOFTWARE DE RECONOCIMIENTO DE DÓLARES AMERICANOS PARA NO VIDENTES .....</b>	<b>131</b>
5.5.1	Análisis de requerimientos .....	131
5.5.2	Diseño de la aplicación .....	132
<b>CAPITULO VI .....</b>	<b>134</b>	
<b>DESARROLLO E IMPLEMENTACION DEL SOFTWARE DE RECONOCIMIENTO DE LA DENOMIAACIÓN DE DOLARES AMERICANOS .....</b>	<b>134</b>	
6.1	INTRODUCCIÓN .....	134
6.2	<b>DESARROLLO DEL SOFTWARE DE RECONOCIMIENTO DE LA DENOMINACIÓN DE DÓLARES AMERICANOS .....</b>	<b>134</b>
6.2.1	Desarrollo de la Etapa de Aprendizaje .....	134
6.2.2	Desarrollo de la etapa de Reconocimiento .....	138
6.3	<b>IMPLEMENTACION DEL SOFTWARE DE RECONOCIMIENTO DE LA DENOMICACIÓN DE DOLARES AMERICANOS .....</b>	<b>142</b>
6.3.1	Requerimientos de Hardware.....	142
6.3.2	Requerimientos de Software .....	142
6.3.3	Instalación del programa en el dispositivo móvil inteligente con Android....	142
6.4	<b>PRUEBAS Y ANALISIS DE DESEMPEÑO .....</b>	<b>143</b>
6.4.1	Pruebas de Reconocimiento .....	143
<b>CAPITULO VII.....</b>	<b>148</b>	
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>148</b>	
7.1	CONCLUSIONES .....	148
7.2	RECOMENDACIONES .....	149
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>150</b>	

## ÍNDICE DE TABLAS

Tabla. 1.1. Teléfonos celulares para personas con discapacidad visual. ....	4
Tabla. 2.2. Herramientas de identificación y ubicación de un Smartphone. ....	13
Tabla. 2.3. Requisitos básicos de herramientas principales de un Smartphone. ....	14
Tabla. 2.4. Requisitos básicos de herramientas opcionales de un Smartphone. ....	15
Tabla. 2.5. Sistema Operativo Android. ....	19
Tabla. 2.6. Sistema Operativo iOS. ....	19
Tabla. 2.7. Sistema Operativo BlackBerry. ....	20
Tabla. 2.8. Sistema Operativo Symbian. ....	20
Tabla. 2.9. Sistema Operativo Windows Mobile. ....	21
Tabla. 2.10. Comparación de detalles básicos entre los SO de los Smartphones. ....	22
Tabla. 2.11. Comparación de la interfaz de usuario entre los SO de los Smartphones. .	23
Tabla. 2.12. Comparación del funcionamiento entre los SO de los Smartphones. ....	24
Tabla. 2.13. Comparación de desarrollo de terceros entre los SO de los Smartphones. .	25
Tabla. 3.1. Distribución de las Versiones de Android (Septiembre del 2011). ....	42
Tabla. 3.2. API Level. ....	44
Tabla. 4.1. Plataformas que soportan Herramientas de Desarrollo para Android. ....	47
Tabla. 4.2. Comparación de comandos de ViewAndroid y View Swing. ....	57
Tabla. 4.3. Lista de Recursos y su Dirección. ....	66
Tabla. 4.4. Subcarpetas de /res/. ....	82
Tabla. 4.5. Propiedades de Table Layout. ....	89
Tabla. 4.6. Acciones de Intents. ....	91
Tabla. 4.7. Etiquetas de Category. ....	93
Tabla. 4.8. Atributos de Datos. ....	94
Tabla. 4.9. Formatos de Audio y Video que admite Android 2.1. ....	98
Tabla. 4.10. Parámetros de la Cámara. ....	103
Tabla. 4.11. Actividades de la Cámara. ....	103
Tabla. 4.12. Comandos opcionales de control de audio. ....	106
Tabla. 6.1. Obtención del Banco de Imágenes de dólares Americanos. ....	135
Tabla. 6.2. Proceso de obtención de Componentes Principales. ....	137
Tabla. 6.3. Configuración de los Parámetros de la Cámara. ....	139

## INDICE DE FIGURAS

Figura. 1.1. Personas con discapacidades registradas y carnetizadas (1996 al 2011). .....	3
Figura. 2.1. Puntos finales de Comunicaciones Unificadas. ....	9
Figura. 2.2. Líneas Móviles y Usuarios de Teléfonos Inteligentes en América Latina..	17
Figura. 3.1. Android SO en Teléfonos Móviles Inteligentes. ....	28
Figura. 3.2. Arquitectura del Sistema Operativo Android.....	29
Figura. 3.3. Conversión del archivo .java a .dex. ....	33
Figura. 3.4. Logo Oficial de Android 1.5 (Cupcake). ....	35
Figura. 3.5. Logo Oficial de Android 1.6 (Donut). ....	36
Figura. 3.6. Logo Oficial de Android 2.0 / 2.1 (Eclair).....	37
Figura. 3.7. Logo Oficial de Android 2.2 (Froyo).....	38
Figura. 3.8. Logo Oficial de Android 2.3 (Gingerbread). ....	40
Figura. 3.9. Logos No Oficiales de las versiones Android 3.X.....	41
Figura. 3.10. Distribución de las Versiones de Android (Septiembre del 2011).....	42
Figura. 4.1. Perspectiva DDMS dentro de Eclipse con el ADT plug-in instalado. ....	50
Figura. 4.2. Ventana configuración de sitio de actualización.....	51
Figura. 4.3. Ventana de Preferencias. ....	52
Figura. 4.4. Android UI – Estructura de Árbol.....	57
Figura. 4.5. Android Manifest Overview. ....	60
Figura. 4.6. Android Manifest Application. ....	60
Figura. 4.7. Carpetas y subcarpetas de Recursos de las Aplicación.....	61
Figura. 4.8. Árbol de prioridad de Procesos. ....	68
Figura. 4.9. Ventana de Nuevo AVD. ....	71
Figura. 4.10. Android SDK y AVD Manager –Instalación de Paquetes. ....	73
Figura. 4.11. Casillas de Paquetes disponibles.....	73
Figura. 4.12. LogCat.....	75
Figura. 4.13. Primera Aplicación Android. ....	77
Figura. 4.14. Ventana de Emulador. ....	79
Figura. 4.15. Run-Dialog.....	80
Figura. 4.16. Run Configuration.....	80

Figura. 4.17. Primera Android Application.....	81
Figura. 4.18. Elementos creados inicialmente para un nuevo proyecto Android.....	81
Figura. 4.19. Subcarpetas de /src/.....	82
Figura. 4.20. Recursos de la Aplicación.....	83
Figura. 4.21. Elementos de código generado automáticamente.....	83
Figura. 4.22. Frame Layout.....	87
Figura. 4.23. Linear Layout.....	88
Figura. 4.24. Relative Layout.....	90
Figura. 4.25. Intents y SubActivities.....	91
Figura. 4.26. Hilos.....	95
Figura. 4.27. Reconocedor de Voz de Android.....	105
Figura. 5.1. Sistema Básico de Reconocimiento.....	109
Figura. 5.2. Modelo de color RGB:.....	112
Figura. 5.3. Imagen representada en el modelo YCbCr.....	114
Figura. 5.4. Imagen de 4:4:4.....	115
Figura. 5.5. Imagen de 4:2:2.....	115
Figura. 5.6. Imagen de 4:2:0.....	116
Figura. 5.7. Tipos de Histograma.....	117
Figura. 5.8. Compresión del Histograma.....	119
Figura. 5.9. Expansión del Histograma.....	120
Figura. 5.10. Desplazamiento del Histograma.....	121
Figura. 5.11. Submuestreo de Imágenes.....	122
Figura. 5.12. Ventaneado de Imagen.....	123
Figura. 5.13. Comparación entre submuestreo por media y por mediana.....	124
Figura. 5.14. Mascara de Filtrado por media.....	125
Figura. 5.15. Filtro menos Laplaciano.....	127
Figura. 5.16. Bloques de la Etapa de Aprendizaje.....	133
Figura. 5.17. Bloques de la Etapa de Reconocimiento.....	133
Figura. 6.1. Obtención del Banco de Imágenes de Dólares Americanos.....	135
Figura. 6.2. Pre procesamiento de Imagen de muestra.....	136
Figura. 6.3. Distancia entre el celular y el billete para el reconocimiento.....	143
Figura. 6.4. Angulo de posición del billete con respecto al celular.....	145

## GLOSARIO

<b>Megapixel</b>	Equivale a 1 millón de píxeles, a diferencia de otras medidas usadas en la computación en donde se suele utilizar la base de 1024 para los prefijos, en lugar de 1000, debido a su conveniencia respecto del uso del sistema binario. Usualmente se utiliza esta unidad para expresar la resolución de imagen de cámaras digitales
<b>Android</b>	Es un sistema operativo basado en el núcleo Linux diseñado originalmente para dispositivos móviles, tales como teléfonos inteligentes
<b>SDK</b>	Es un conjunto de herramientas y programas de desarrollo que permite al programador crear aplicaciones para un determinado paquete de software, estructura de software, plataforma de hardware
<b>Eclipse IDE</b>	Eclipse es principalmente una plataforma de programación, usada para crear entornos integrados de desarrollo (del Inglés IDE).
<b>Java</b>	Es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems, permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios
<b>AVD</b>	Un dispositivo Android Virtual (AVD) es una configuración del emulador que permite modelar un dispositivo real mediante la definición de opciones de hardware y software

<b>API</b>	Interfaz de Programación de Aplicaciones, es el conjunto de funciones y procedimientos o métodos, en la programación orientada a objetos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
<b>Dalvik</b>	Dalvik es una máquina virtual intérprete que ejecuta archivos en el formato Dalvik Executable (*.dex), un formato optimizado para el almacenamiento eficiente y ejecución mapeable en memoria.
<b>JDK</b>	Es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red.
<b>ADT</b>	Herramientas de desarrollo de Android (ADT) es un plug-in para el IDE de Eclipse que está diseñado para darle un ambiente potente, integrado en el que la creación de aplicaciones para Android
<b>Plug-in</b>	Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.
<b>Widgets</b>	Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de Widget Engine. Entre sus objetivos está dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.
<b>GUI</b>	Interfaz Gráfica de Usuario, es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz
<b>Memoria Caché</b>	Es aquella que acelera los procesos repetitivos del sistema, es mucho más veloz que la RAM y está en contacto directo con el CPU.

<b>Shell</b>	Es un programa que lee los comandos que se teclean y los convierte en una forma más entendible para el sistema Unix/Linux. También incluye algunas sentencias básicas de programación que permiten: tomar decisiones, realizar ciclos y almacenar valores en variables.
<b>Trackball</b>	Es un dispositivo apuntador estacionario compuesto por una bola incrustada en un receptáculo que contiene sensores que detectan la rotación de la bola en dos ejes.
<b>ADB</b>	Android Depuración Bridge, es una herramienta de línea de comandos versátil que le permite comunicarse con una instancia de emulador o un dispositivo conectado Android.
<b>SQLite</b>	Es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca en C++.
<b>URI</b>	Identificador Uniforme de Recursos, permite incluir en la dirección una subdirección, determinada por el fragmento.
<b>Teorema de Bayes</b>	Vincula la probabilidad de A dado B con la probabilidad de B dado A.
<b>Pixel</b>	Es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.
<b>RGB</b>	Representa una composición del color en términos de los colores primarios con los que se forma (rojo, verde y azul). Está basado en la síntesis aditiva, es decir la mezcla por adición de los tres colores luz primarios.
<b>YCbCr</b>	Es un espacio de color utilizado en sistemas de fotografía y video digital. Define el color en términos de un componente de luminancia y dos de crominancia (roja y azul).

<b>YUV</b>	Es un espacio de color usado como parte de un conducto de imagen en color. Codifica una imagen o video en color teniendo en cuenta la percepción humana, permite utilizar ancho de banda reducido. para los componentes de crominancia, de esta forma, hace que los errores de transmisión o las imperfecciones de compresión se oculten más eficientemente a la percepción humana.
<b>Submuestreo</b>	Es generar una imagen a partir de tomar muestras periódicas de la imagen original, de tal forma que esta quede más pequeña en tamaño o dimensión.
<b>Croma</b>	Es una técnica audiovisual utilizada ampliamente tanto en cine y televisión como en fotografía, que consiste en la sustitución de un fondo por otro mediante un equipo especializado o un ordenador.
<b>Bit</b>	El bit es la unidad mínima de información empleada en informática, en cualquier dispositivo digital, o en la teoría de la información.
<b>Histograma</b>	Representación gráfica de la distribución que existe de las distintas tonalidades de grises con relación al número de pixeles o porcentaje de los mismos.
<b>Binarización</b>	Consiste en un proceso de reducción de la información de la misma, en la que solo persisten dos valores: verdadero y falso. En una imagen digital, estos valores, verdadero y falso, pueden representarse por los valores 0 y 1.
<b>Superficie Lambertiana</b>	Es aquella que refleja la energía uniformemente hacia todas las direcciones.
<b>Filtro Laplaciano</b>	Es una medida 2-D isotrópica de la 2nd derivada espacial de una imagen.

<b>Eigenvalues</b>	Valores propios, son un conjunto especial de escalares asociada a un sistema de ecuaciones lineales (ecuación de la matriz) que a veces también es conocido como raíces características o valores característicos.
<b>Eigenvectors</b>	Vectores propios, son los no-cero vectores que, después de haber sido multiplicada por la matriz, ya sea siendo proporcional al vector original llegar a ser cero.
<b>Eigenfaces</b>	Es un conjunto de detalles de caras estandarizados, derivado de un análisis estadístico de muchas imágenes de rostros.
<b>Distancia Euclideana</b>	Es la distancia ordinaria entre dos puntos de un espacio Euclídeo, Índice cuantitativo que mide la separación existente entre dos unidades de observación según los valores que ellas posean en un conjunto de variables
<b>PCA</b>	Análisis de Componentes Principales, es un procedimiento matemático que utiliza una transformación ortogonal para convertir un conjunto de observaciones de las variables posiblemente correlacionadas en un conjunto de valores de variables no correlacionadas llamadas componentes principales.
<b>Frame</b>	Es una imagen particular dentro de una sucesión de imágenes que componen una animación. La continua sucesión de estos fotogramas producen a la vista la sensación de movimiento, fenómeno dado por las pequeñas diferencias que hay entre cada uno de ellos.

## **CAPITULO I**

### **PERSONAS CON DISCAPACIDAD VISUAL**

#### **1.1 INTRODUCCIÓN**

En el Ecuador existen aproximadamente 313 mil personas con discapacidad, de los cuales 35 mil personas poseen discapacidad visual, concentrando mayor incidencia en las provincias de Guayas, Manabí y Pichincha según información obtenida del Consejo Nacional de Discapacidades (CONADIS), en el periodo de 1996 – 2011. Este grupo de personas se desenvuelve en la vida con cierta desventaja, sin embargo ellos han desarrollado el resto de sus sentidos para valerse por sí mismos según la Federación Nacional de Ciegos (FENCE). Aun así no es suficiente para realizar sus actividades con normalidad.

##### **1.1.1 Población no vidente en el Ecuador**

Ecuador gracias a la gestión de la Vicepresidencia a cargo del Lic. Lenin Moreno, ha logrado ejecutar “La Ley Orgánica de Discapacidades, el Primer Plan Nacional de Discapacidades y el establecimiento en el Reglamento a la Ley de las competencias, responsabilidades y atribuciones que tienen las distintas instituciones del sector público y privado en la prevención, atención e integración, así como la obligatoria necesidad de coordinación y participación de las mismas, el fortalecimiento de las organizaciones de personas con discapacidad y la creación de la Red de ONG's” [1].

En este proceso de equiparación de derechos, los organismos de gobierno e independientes han enfocado su ayuda desde otra perspectiva, y es la de ayudar a las personas con discapacidades no por caridad, si no por brindar una ayuda eficiente de rehabilitación y atención personalizada, priorizando su estado de salud y posteriormente su calidad de vida en conjunto con el Plan del Buen Vivir citado en el anexo A1.

Una de las gestiones más importantes de la Vicepresidencia es el de incursionar a personas con discapacidad al ámbito laboral como se menciona en el objetivo 10.3 del Plan del Buen Vivir, señalando que la integración de las personas con discapacidad sea una verdadera inclusión donde puedan ejercer sus derechos ciudadanos y su cumplimiento laboral, este objetivo se lo puede ver en el anexo A2. “La meta debe ser eliminar barreras y hacer que el mundo de la comunicación, la interacción y la cooperación sea accesible para todos ya que una de las maneras para evaluar una sociedad humana es a través de las oportunidades que ofrece a sus individuos más débiles”<sup>1</sup>.

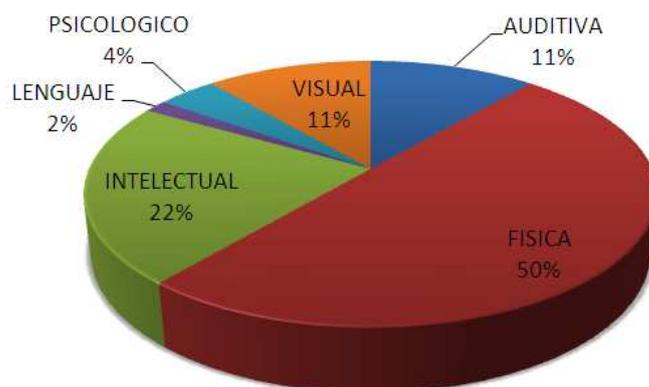
### 1.1.2 Estadísticas del CONADIS

Existen gran cantidad de personas descritas como discapacitadas en el Ecuador pero que no están registradas, calificadas o examinadas, ni han sido oficialmente declaradas como “persona con discapacidad”. Es por eso que existe un consejo encargado del proceso de calificación y carnetización a las personas con discapacidad llamado CONADIS a nivel nacional; su función consiste en registrar a las personas con discapacidad en el Registro Nacional de Discapacidades y así estas puedan obtener el carnet de discapacidad, lo cual le permite acceder a la condición legal de “persona con *discapacidad*” y mediante el cual se puede tener acceso a toda la ayuda y control del gobierno [1].

A continuación se muestra en la Figura 1.1 la información de las personas carnetizadas y registradas en el CONADIS desde el año 1996 hasta la fecha. Se tomó este dato ya que es la información más acertada en la cifra total de las personas con discapacidad en el Ecuador. La discapacidad visual ocupa el tercer lugar, siendo el 11% del total de las personas con discapacidad a nivel nacional [1].

---

<sup>1</sup>Battro, Denham, 2004



**Figura. 1.1. Personas con discapacidades registradas y carnetizadas (1996 al 2011).**

## 1.2 EL ROL DE LA TECNOLOGÍA

Las investigaciones tecnológicas se han desarrollado a medida que las necesidades y requerimientos de las personas han ido apareciendo, y en el caso de aquellas con discapacidad visual las necesidades son muchas como: identificar colores, detalles de los objetos, características de las personas entre otras.

Las tecnologías deben ayudar a facilitar la calidad de vida de las personas incluidas personas con discapacidad, proporcionando la capacidad de desarrollar y mejorar sus habilidades habituales, permitiéndoles realizar actividades que con anterioridad eran irrealizables. En especial la tecnología celular, representada por los dispositivos móviles son la opción tecnológica más apropiada para brindar ayuda a las personas con discapacidad, ya que por sus grandes características como: movilidad, relativa facilidad de uso y acceso, se ha convertido en el equipo más utilizado y necesitado a nivel mundial. “La demora en el desarrollo de herramientas para personas con discapacidad no es debida a la falta de tecnología, sino al escaso reconocimiento de su potencial”<sup>2</sup>.

---

<sup>2</sup> Levitt, 1982

### 1.2.1 Tecnología Celular en beneficio de las personas no videntes

La tecnología se ha enfocado a proyectos sociales, y es por eso que varias empresas como Samsung, HTC y Sony Ericsson están diseñando teléfonos móviles inteligentes con aplicaciones y modos para no videntes con comandos por voz y señales al tacto, utilizando como base el sistema operativo Android desarrollado por Google, y otros pero sin mucho aporte, los cuales brindan facilidades de uso a las personas con discapacidad visual.

A continuación se muestra en la Tabla 1.1 un resumen de modelos más relevantes de dispositivos móviles diseñados para personas con discapacidad visual, son modelos exclusivos que se realizan bajo pedidos de instituciones dedicadas a brindar ayuda a personas discapacitadas, y a un módico precio.

**Tabla. 1.1. Teléfonos celulares para personas con discapacidad visual.**

Modelo	Descripción	Imagen
<b>B-Touch</b>	Tiene interfaz táctil desarrollada por Google, el cual dispone de una pantalla celular para ciegos táctil-braille, es una matriz de puntos que muestra todo en lenguaje braille. El sistema táctil se complementa con un sistema de reconocimiento de voz y de caracteres [2].	
<b>Sens</b>	Concept Phone, de la marca diseñadora Takumi Yoshida, agrega teclas de un tamaño más grande en donde cada tecla está conformada por un relieve al tacto [3]. Posee un sistema de avisos por voz para realizar las acciones deseadas como hacer llamada, mandar mensajes de texto.	
<b>FayerWayer</b>	Dispositivo de la marca Samsung, cambia su interfaz táctil-braille de acuerdo a la función deseada, transforma los mensajes de texto de cualquier idioma a braille y viceversa. Gracias a la tecnología EAP <sup>3</sup> el equipo modifica el contenido de su pantalla, haciendo que los filamentos plásticos, formados en matrices de 2 por 3, suban y bajen [4].	

<sup>3</sup>EAP: Electric Active Plastic

<p><b>Braille Reader</b></p>	<p>Es una aplicación Nokia, que ayuda a las personas con poca visión o nula, a leer los mensajes de texto. Lo que hace esta aplicación es convertir los mensajes de texto en vibraciones en la pantalla que una persona no vidente y con conocimientos de Braille fácilmente podrá descifrar [5].</p>	
------------------------------	---	---

La tecnología celular aunque ha hecho algunos avances para brindar soluciones a las personas invidentes, aún queda poder desarrollar una solución concreta. Debido a la falta de soluciones este proyecto de identificación de la denominación de billetes se encamina a ser una potencial solución para las personas que tiene la dificultad de identificar el valor del billete.

### 1.3 IDENTIFICACIÓN DE OBJETOS DE PERSONAS NO VIDENTES

Uno de los principales inconvenientes de este grupo de personas es el de identificar cosas, y para ello se han inventado muchos métodos como es el sistema Braille siendo el más utilizado ya que les permite leer mediante puntos en superficie con relieve.

De igual forma existen muchos otros métodos que les ayudan a identificar objetos implementados en teclados, pantallas, libros, cámaras fotográficas entre otras, permitiéndoles realizar actividades vinculadas con el mundo tecnológico. Sin embargo en el ámbito monetario no se ha dado una solución, impidiéndoles realizar sus transacciones monetarias de manera personal. Es por ello que se pretende con este proyecto dar una solución a la identificación de la denominación de billetes dirigido a personas no videntes.

#### 1.3.1 Identificación de la denominación de billetes

Dentro de las necesidades más importantes es la identificación de la denominación de dólares americanos que es la moneda circulante en el Ecuador, los invidentes utilizan métodos no técnicos inventados por ellos para reconocer los billetes como es doblar las esquinas, colocar una señal al tacto, o simplemente requieren de una persona que les ayude a identificarlos y realizar sus transacciones monetarias.

### 1.3.2 Aplicación celular de identificación de billetes basada en Android SO

Debido a que el celular tiene características deseables como es un entorno amigable para su utilización, fácil transporte, y relativa facilidad de adquisición del dispositivo, facilita la implementación del software de reconocimiento de Dólares Americanos, que puedan utilizar las personas con discapacidad visual.

Este software estará encaminado a funcionar bajo el sistema operativo para móviles inteligentes lanzada por Google llamada Android, esto debido a que esta plataforma tiene grandes posibilidades de convertirse en el sistema operativo dominante para teléfonos inteligentes ya que el 50% de dispositivos viene con el sistema operativo Android según datos de la Open Handset Alliance<sup>4</sup>.

Android SO tiene el objetivo de ser una de las primeras plataformas completamente independientes de la visión. Ello es debido a que soporta vibraciones y sonidos, además porque sus dispositivos disponen de sensores de movimiento, proximidad, equilibrio y luz por lo que es posible agregar la opción invidente a las aplicaciones de este software.

### 1.4 ALCANCE DEL PROYECTO

Al finalizar el proyecto propuesto se quiere obtener como producto final, un software de reconocimiento de denominación de Dólares Americanos a través de la cámara de un teléfono móvil inteligente con sistema operativo Android. Para esto se considera los siguientes requisitos importantes.

#### **Acierto**

Lo más importante del software es que tenga un alto porcentaje de acierto en cuanto al reconocimiento de la denominación del billete, se logrará este porcentaje de aciertos tomando en cuenta todos los diferentes formatos de billetes de todas las denominaciones, e incluso se considera el estado del billete: viejo o nuevo. Se utilizará

---

<sup>4</sup>Open Handset Alliance: Es una alianza comercial de 84 compañías para desarrollar estándares abiertos para dispositivos móviles.

algoritmos de Reconocimientos de Patrones, identificando el rostro impreso en el billete que será comparado con un banco de imágenes antes adquiridas de todos los rostros de cada denominación en todas sus versiones.

### **Velocidad de Respuesta y Presentación de resultados**

El software tendrá una mayor velocidad en respuesta, teniendo un aceptable tiempo que oscile entre 2 y 3 segundos, siendo un tiempo aceptable para el reconocimiento de un billete. Además preste mayor facilidad de utilización e interactividad con la persona no vidente. Se lograra este requisito optimizando los recursos de memoria y código del software.

### **Condiciones de Luz y Posición**

La aplicación funcionará en todas las condiciones de luz existentes, bien sea en luz natural o artificial. Para lograr este requisito se utilizara algoritmos de Procesamiento Digital de Imagen que se verá en el Capítulo 5. Estos algoritmos nos permiten influenciar en la calidad de la imagen obtenida y posición para que siempre la imagen entre a un proceso de análisis y reconocimiento sin tener mucha información inservible como sombras, manchas o caracteres no definidos.

### **Interactividad y Notificaciones por voz**

Utilizando las herramientas de audio y pantalla capacitiva que nos brinda el sistema operativo Android y el dispositivo móvil inteligente en el cual será instalado la aplicación, se convertirá el software de reconocimiento en una aplicación con funciones de señales de voz y vibración para que tenga una mayor interactividad con el usuario no vidente. Cuando el software tenga alguna notificación ya sea de respuesta o instrucción, enviará una señal de voz al usuario.

## **CAPITULO II**

### **TELÉFONOS MÓVILES INTELIGENTES**

#### **2.1 GENERALIDADES**

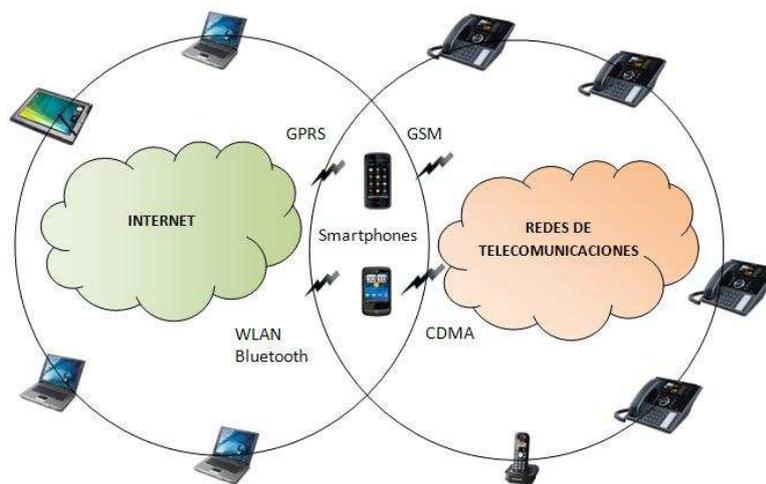
El incremento del mercado de la telefonía móvil en América latina y el desarrollo tecnológico a nivel mundial, aumenta la penetración de la tecnología móvil y del internet en dispositivos móviles, siendo el aparato más utilizado en el mundo. Siguiendo esta tendencia, el mercado de teléfonos móviles inteligentes en América Latina está creciendo a un ritmo de tres dígitos anuales con proyecciones de alcanzar los 130 millones de unidades en toda la región para el año 2014, estadísticamente hablando según la consultora Gartner. Este indicador representa aproximadamente un teléfono inteligente por cada cuatro usuarios de telefonía móvil en América Latina [6].

Este fenómeno se debe también al incremento en el desarrollo de nuevas ofertas de servicios móviles dirigidos al público de negocios, jóvenes, adolescentes e incluso niños siendo más del 80% el público de negocios y universitarios los que más usan estos dispositivos, y es que la inclusión rápida de las redes sociales han ayudado a incrementar las ventas de teléfonos móviles inteligentes en todo el mundo. Estos nuevos servicios ayudan a realizar actividades a través del internet, ya sean: pagos, compras, transferencias e inclusive consultas al médico. De cierta forma brindan unificación de servicios en un solo dispositivo, sin embargo en el Ecuador existe mucha gente que prefiere hacer sus actividades de la manera convencional por cuestiones de seguridad o simple desconocimiento.

El Ecuador aun es un país que adopta tecnologías mas no las desarrolla en su totalidad, sin embargo con la investigación y el conocimiento adecuado se puede empezar a desarrollar aplicaciones para dispositivos móviles en Ecuador, como es el caso de este proyecto de desarrollo de una aplicación para dispositivos celulares con Android SO.

## 2.2 TELÉFONOS MÓVILES INTELIGENTES

Los teléfonos móviles inteligentes son la tendencia de las comunicaciones unificadas, integra los servicios de telecomunicaciones e internet en un solo dispositivo, ya que ha combinado la portabilidad de los teléfonos móviles con la potencia informática y las redes de telecomunicaciones [7], como se ilustra en la Figura 2.1.



**Figura. 2.1. Puntos finales de Comunicaciones Unificadas.**

Debido a que los teléfonos móviles inteligentes suelen ser tan poderoso como un PC de pocos años atrás, sus sistemas operativos han evolucionado para ser más bien de características propias móviles. Los sistemas operativos de teléfonos móviles inteligentes en la actualidad son Symbian OS, Windows Mobile OS, Apple con iOS, Research in Motion con BlackBerry OS y Android OS, estos serán analizados en la sección 2.5.

### 2.2.1 Evolución Histórica

En la Tabla 2.1 se observa una breve historia de la evolución de los teléfonos móviles inteligentes, desde sus comienzos con equipos monocromáticos hasta la tecnología de hoy en día, capaz de reproducir y capturar video en alta definición, entre muchas otras aplicaciones.

**Tabla. 2.1. Reseña histórica de teléfonos móviles inteligentes.**

Modelo	Descripción	Imagen
<b>Simon de IBM (1993)</b>	Fue el primer intento real de la industria tecnológica de incorporación de servicios de voz y datos, por lo que el equipo funcionaba como un teléfono móvil, asistente digital personal, con correo electrónico e incluso como una máquina de fax y teclado QWERY [8].	
<b>Palm Pilot (1996)</b>	No fue técnicamente un teléfono inteligente, fue el equipo que popularizó la sigla PDA en los Estados Unidos. Contaba con un procesador de 16MHz y una memoria de 128KB [8].	
<b>Nokia 9110 Communicator (1998)</b>	El Nokia 9110 Communicator fue un dispositivo con un diseño más similar a lo que hoy entendemos como Smartphone. Su pantalla no era color, y no se podía navegar por Internet, pero tenía un teclado QWERTY deslizable que sirvió como modelo para los teléfonos actuales [8].	
<b>BlackBerry 5810 (2002)</b>	Research In Motion (RIM) era conocida gracias a sus beepers <sup>5</sup> . Pero a comienzos de 2002, RIM entró en el mercado de los teléfonos móviles. Su BlackBerry 5810 era un teléfono con la capacidad de revisar correos electrónicos y navegar por Internet. Su desventaja era que requería usar un auricular para poder hablar [8].	
<b>Palm Treo 600 (2003)</b>	El Treo 600 fue el primer Smartphone lanzado por Palm, tras adquirir al fabricante Handspring. Este móvil tenía la particularidad de soportar redes GSM y CDMA, tenía 32MB de memoria RAM y un procesador de 144MHz. Fue un equipo que se vendió muy bien, aunque fue lanzado en 2003, una época en la que Palm comenzaba su caída en popularidad [8].	

<sup>5</sup> Beeper: Dispositivo de telecomunicaciones muy simple que recibe mensajes de texto corto.

<p><b>iPhone de Apple (2007)</b></p>	<p>Presenta una inusual pantalla táctil con gran resolución, y ofrecía la mejor experiencia en Internet hasta ese momento. Han pasado 3 años, y todavía el iPhone es el Smartphone con el que los demás equipos son comparados [8].</p>	
<p><b>Droid de Motorola (2009)</b></p>	<p>Google presenta Android: En el mismo año en que Apple lanzó el iPhone. Si bien Android ya tenía su buen tiempo en el mercado, el Droid de Motorola fue el primer móvil súper exitoso en utilizar la plataforma Android en los Estados Unidos [7].</p>	
<p><b>HTC EVO 3D (2011)</b></p>	<p>Es un Smartphone con Android OS que posee una pantalla de 4.3 pulgadas con resolución de 540 x 960 pixels (qHD) con posibilidad de visualizar contenido 3D, imágenes y video, sin tener que usar gafas. Además, posee un procesador Snapdragon de doble núcleo a 1.2GHz y memoria RAM de 1GHz, cámara de 5 megapíxeles con captura de video 3D en HD y corre Android 2.3 Gingerbread [8].</p>	

Los teléfonos móviles inteligentes se les otorga ese calificativo debido a que en su historia eran simples dispositivos que realizaban llamadas, leían texto y en el mejor de los casos podían leer correos y enviar fax. Hoy en día un teléfono móvil inteligente puede desarrollar más de mil funciones personalizables a las necesidades del usuario desde configurarlo como un despertador, planificar todas las citas del día, planificar su ruta de viaje, observar tiendas y productos disponibles en el mercado, comprar, vender, ofertar, hasta trabajar en cualquier parte del mundo como si se estuviera en su propia oficina, muy similar a un computador portátil e incluso con mayores funciones. Es un dispositivo bastante configurable para controlar y planificar un día de trabajo cotidiano.

### 2.3 CARACTERÍSTICAS DE LOS TELÉFONOS MÓVILES INTELIGENTES

A continuación se muestra un recuento de las características básicas que tienen los teléfonos móviles inteligentes según fabricantes de estos dispositivos.

### 2.3.1 Memoria interna

Los teléfonos móviles inteligentes ofrecen a los usuarios una mejora de las aplicaciones PIM<sup>6</sup>, la capacidad de enviar y recibir correo electrónico, conectarse a Internet, realizar y recibir llamadas. Los datos se mantienen en dos regiones: memoria de sólo lectura ROM y memoria de acceso aleatorio RAM. Típicamente, el sistema operativo y aplicaciones pre cargadas suministradas por el fabricante se almacenan en la memoria ROM que proporciona protección contra el borrado durante el caso de un restablecimiento completo o el agotamiento de la batería. La memoria RAM se divide en dos regiones, la memoria del programa y un almacén de objetos [9].

- **Memoria de programa:** Se usa para la ejecución del programa, los conductores de carga y almacenamiento para procesar la información. Esta memoria se borra al igual que la memoria RAM de un ordenador personal.
- **Almacén de objetos:** Conserva los datos durante los estados activos y en reposo, pero se pierden los datos en caso de agotamiento de la batería o un restablecimiento completo.

Los fabricantes ofrecen a los usuarios de dispositivos inteligentes una carpeta de seguros asignada por defecto, proporcionando la capacidad de proteger el borrado de datos definidos en el caso de un restablecimiento completo o agotamiento de la batería. Aunque los datos presentes en los teléfonos inteligentes se pueden almacenar en un formato propio.

### 2.3.2 Tarjeta SIM

Debido a la norma GSM 11.111, de dispositivos móviles diseñados para extraer datos de una tarjeta SIM<sup>7</sup> ya sea interna o externa con un lector de tarjetas SIM, debe ser capaz de adquirir correctamente, decodificar y presentar datos en un formato legible para el usuario. Una abundancia de información se almacena en la tarjeta SIM, como

---

<sup>6</sup> PIM (Personal Information Management)

<sup>7</sup> Tarjeta SIM (Subscriber Identity Module): Tarjeta pequeña y desmontable que puede ser insertada en un teléfono móvil con el fin de proporcionar servicio con una compañía de telefonía móvil en particular, llamada comúnmente en Ecuador como Chip.

números de marcación abreviada (ADNS), últimos números marcados (LND), mensajes (SMS), información sobre los abonados y la información de ubicación (LOCI) y el General Packet Radio de Ubicación del servicio (GPRSLOCI) [9].

### 2.3.3 Evidencia Digital

La cantidad y riqueza de datos contenidos en los teléfonos móviles inteligentes puede variar dependiendo del fabricante y el sistema operativo. Aplicaciones pre cargadas y la posibilidad de instalar aplicaciones personalizadas proporcionan a los usuarios soluciones sin fin [9]. Las herramientas deben tener la capacidad de cubrir los siguientes objetos de datos almacenados en la memoria del dispositivo portátil interno y los medios asociados, que se pueden observar en la Tabla 2.2.

**Tabla. 2.2. Herramientas de identificación y ubicación de un Smartphone.**

Herramientas de identificación	Función	Ubicación
IMEI	Identificador Internacional de Equipo Móvil	Memoria Interna
MEID	Identificador de equipo móvil	Memoria Interna
ESN	Número de Serie Electrónico	Memoria Interna
SPN	Proveedor de Servicios de Nombre	Memoria SIM
IMSI	IMSI: Identidad internacional del abonado móvil	Memoria SIM
MSISDN	Abonado móvil internacional número RDSI <sup>8</sup>	Memoria SIM
PIM	Gestión de información personal (libreta de direcciones, entradas de agenda, lista de tareas, Tareas, Memos)	Memoria Interna
ADNS	Números de Marcación abreviada	Memoria SIM
LND	Los números marcados	Memoria SIM
SMS, EMS	Mensajes de texto	Memoria SIM/ Memoria Interna
MMS	Mensajes Multimedia (correo electrónico y los datos asociados audio, gráficos, vídeo)	Memoria Interna
IFS	Almacenamiento de Archivos Independientes (imágenes, audio y video)	Memoria SIM/ Memoria Interna
LOCI	Información de ubicación	Memoria SIM
GPRSLOCI	General Packet Radio de Ubicación del servicio.	Memoria SIM

<sup>8</sup>RDSI: Red Digital de Servicios Integrados

### 2.3.4 Requisitos

Los requisitos de herramientas de un teléfono móvil inteligente se pueden clasificar en dos secciones: Smart Phone Tool-Core: obligatorio-01 (SPT-CR-01) a través de tubos sin soldadura CR-01-06 y Smart Phone Tool-requisito: opcional-01 (SPT-RO-01) a través de tubos sin soldadura RO-01-15 [9].

#### Requisitos para las características principales

Los requisitos básicos siguientes vistos en la Tabla 2.3, correrán a cargo de todas las herramientas principales de dispositivos móviles capaces de adquirir la memoria interna del teléfono inteligente [9].

**Tabla. 2.3.Requisitos básicos de herramientas principales de un Smartphone.**

Herramientas principales	Código	Función
<b>Reconocimiento</b>	SPT-CR-01	Reconoce los dispositivos compatibles a través del proveedor de interfaces de apoyo (por ejemplo, por cable, Bluetooth, infrarrojos).
	SPT-CR-02	Identificar los dispositivos que no reciben apoyo.
<b>Notificación</b>	SPT-CR-03	Capacidad para notificar al usuario de los errores de conectividad entre el dispositivo y la aplicación durante la adquisición.
<b>Adquisición</b>	SPT-CR-04	Proporcionar al usuario, ya sea con una vista previa panel o ver el informe generado de los datos adquiridos.
	SPT-CR-05	Adquirir lógicamente todas las aplicaciones compatibles objetos de datos presentes en la memoria interna.
	SPT-CR-06	Adquirir lógicamente apoyado objetos de datos sin cambiar los objetos de datos presentes en el dispositivo.

#### Requisitos para las características opcionales

Los siguientes requisitos de herramienta de un teléfono móvil inteligente definen las características opcionales vistas en la Tabla 2.4. Si una herramienta proporciona la capacidad de definir, la herramienta es la prueba de conformidad con estos requisitos,

pero si no proporciona la capacidad de definir, el requisito no se aplica [9]. Las características opcionales identifican lo siguientes:

**Tabla. 2.4. Requisitos básicos de herramientas opcionales de un Smartphone.**

Herramientas opcionales	Código	Función
<b>SIM Adquisición</b>	SPT-RO-01	Reconocer el apoyo a través de tarjetas SIM compatible con el proveedor de interfaz (por ejemplo, lector PC / SC, lector de propiedad, en la residencia).
	SPT-RO-02	Identificar SIM no compatible.
	SPT-RO-03	Notificar al usuario de los errores de conectividad entre el lector SIM y la aplicación durante la adquisición.
	SPT-RO-04	Adquirir todos los datos de apoyo de la aplicación objetos presentes en la memoria SIM.
<b>Presentación</b>	SPT-RO-05/06	Proporcionar una presentación de los datos adquiridos en un formato legible para el usuario a través de un informe generado y una vista del panel de vista previa.
<b>Contraseña SIM Protegidas</b>	SPT-RO-07	Una Proporcionar al usuario la oportunidad de desbloquear una contraseña protegida SIM antes de lector externo de adquisición de SIM.
<b>Integridad de datos</b>	SPT-RO-08	Proteger adquirido previamente los objetos de datos dentro de un expediente guardado de la modificación.
<b>Adquisición Física</b>	SPT-RO-09	Adquisición física de la memoria interna del dispositivo para los dispositivos compatibles.
<b>Caracteres no ASCII</b>	SPT-RO-10	Presentar los objetos de datos que contienen caracteres no ASCII adquirido de la memoria interna del dispositivo o tarjeta SIM a través de la interfaz seleccionada (el panel de vista previa, el informe generado). Los caracteres no-ASCII se imprimirán en su representación nativa.
<b>Los intentos de PIN</b>	PT-RO-11	Presentar el número restante de intentos de desbloquear el PIN CHV1/CHV2.
<b>Los intentos PUK</b>	SPT-RO-12	Presentar el número restante de intentos de desbloquear PUK.
<b>Adquisición independiente</b>	SPT-RO-13	Adquirir datos de memoria interna, sin modificar los datos presentes en la tarjeta SIM.
<b>Hashing</b>	SPT-RO-14	Calcular el hash de objetos de datos individuales.
<b>Coordenadas GPS</b>	SPT-RO-15	Adquirir datos relacionadas con GPS en la memoria interna.

## 2.4 MERCADO DE TELÉFONOS MÓVILES INTELIGENTES

Se proyecta que las ventas de teléfonos móviles inteligentes en América Latina crecerán debido a una mayor oferta de equipos y precios competitivos estimulada por la competencia entre operadores y distribución de equipos de grandes fabricantes que exportan los productos a América Latina [7].

En el año 2009, los usuarios de teléfonos móviles inteligentes en América Latina era de aproximadamente 31 millones de unidades representando aproximadamente el 7,9% del total de abonados móviles en toda la región. Para el año 2011, el número de abonados móviles se estima que crezca un 17% en comparación con el año 2009, alcanzando una cantidad de 459 millones de suscriptores. En este período, el número de usuarios de teléfonos móviles inteligentes en América Latina se proyecta que crecerá en un 128%, llegando a los 70 millones de unidades que representa el 15,3% del total de abonados móviles según la consultora Gartner [7].

Debido a cierta reducción en el precio promedio de estos dispositivos, atractivos planes de datos que ofrecen las operadoras y la necesidad de estos aparatos, más personas han adquirido un teléfono móvil inteligente. Las ganancias de los distribuidores especializados en la venta de dispositivos móviles se triplicarían y, para fines de 2011, alcanzarían los 15.100 millones de dólares. Para el año 2014, los analistas proyectan un aumento de 10 veces en los ingresos de la industria, que alcanzó los 5.200 millones de dólares en 2010 [7].

El 81% de las aplicaciones para dispositivos móviles inteligentes seguirán siendo gratis y esa es la tendencia de Android para el año 2011. Los ingresos obtenidos por las ventas de publicidad crecen gradualmente. Al final del año 2010 representaron aproximadamente el 16% de las ganancias de los distribuidores, a finales del año 2014, según analistas del medio, la publicidad ascenderá a un tercio de los ingresos [7].

El proveedor más grande en tiendas de aplicaciones es App Store, de la corporación Apple, lanzado en 2008. Sus aplicaciones superan los 350.000 programas

para iPhone, iPad e iPod representando el noventa por ciento de descargas de aplicaciones móviles en el año 2010.

El crecimiento en el mercado de aplicaciones para teléfonos móviles inteligentes se debe también a la gran aparición de la tienda en línea Android Market. Que hoy en día está cerca a la tienda de App Store de Apple con 134.342 aplicaciones no gratuitas y con más de 100.000 aplicaciones gratuitas. Los analistas indican que, durante 2011, el sistema operativo Android, la base para los dispositivos HTC, Motorola Mobility, Samsung y otros, podría convertirse en la plataforma más popular para los teléfonos inteligentes del mundo [10].

La tienda en línea Windows Phone Market place, en la actualidad, alberga un total de cerca de 7.000 productos de software para teléfonos inteligentes de la plataforma Microsoft Windows Phone 7 [10].

La ascensión imparables de Android OS en el mercado de sistemas operativos móviles se debe a la imparable caída de Symbian, que fue líder tradicional en sistemas móviles hace pocos años, que ha visto perder prácticamente 20 puntos de cuota de mercado en un año [7]. A continuación observamos la Figura 2.2, donde nos muestra una comparativa del crecimiento de teléfonos móviles inteligentes frente a líneas móviles.

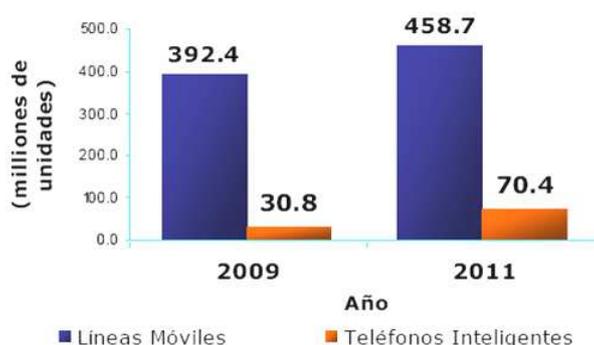


Figura. 2.2. Líneas Móviles y Usuarios de Teléfonos Inteligentes en América Latina.

## 2.5 SISTEMAS OPERATIVOS DE TELÉFONOS MÓVILES INTELIGENTES

Esta sección trata de los Sistemas Operativos para Teléfonos móviles inteligentes, para ello, se incluirá las siguientes preguntas:

¿Qué es un sistema operativo?

Es un conjunto de programas que se integran con el hardware para facilitar al usuario, el aprovechamiento de los recursos disponibles [11]. Algunas de sus características principales son:

- Gobernar y proveer de un ambiente conveniente de trabajo.
- Hacer uso eficiente del Hardware.
- Proveer de una adecuada distribución y asignación de los recursos.
- Administrar y controlar la ejecución de los programas.

¿Qué es un sistema móvil?

Un sistema móvil es un sistema informático que no está vinculado a un determinado lugar. Es posible moverlo o llevarlo a cualquier parte, estos son: un teléfono celular, una computadora de mano o un sistema informático especial en un coche. Aunque existen muchas similitudes entre un fijo y un sistema operativo para móviles, también hay claras distinciones relativas a la movilidad.

A continuación se hace un resumen de las características e información relevante de los diferentes sistemas operativos vigentes en el Ecuador para dispositivos móviles inteligentes.

La información de las siguientes Tablas 2.5- 2.9 [12] permite saber el fabricante y año en el que fue creado el sistema operativo, además de las diversas versiones del sistema operativo desde su inicio hasta la actualidad y su compatibilidad con los equipos telefónicos que acepten su sistema operativo.

## 2.5.1 Android OS

**Tabla. 2.5. Sistema Operativo Android.**

<b>Sistema Operativo</b>	<b>Android</b>
<b>Fabricante</b>	Android Inc. (2008)
<b>Versiones</b>	V.1.5 - V.1.6 - V.2.0 - V.2.1 - V.2.2 - V.2.3 - V.3.0 - V.3.1 - V.3.2
<b>Compatibilidad con Dispositivos</b>	HTC, Sony Ericson, Samsung, LG, Motorola, Huawei
<b>Características Principales</b>	<ul style="list-style-type: none"> <li>▪ Adaptable a pantallas más grandes y multitáctiles, VGA, biblioteca de gráficos 2D, 3D basado en OpenGL ES 2.0.</li> <li>▪ Soporta tecnologías: GSM/EDGE, CDMA, UMTS, Bluetooth, Wi-Fi, LTE y WiMAX.</li> <li>▪ Navegador Web basado en el motor de código abierto WebKit, con el motor JavaScript V8 y SQLite base de datos liviana.</li> <li>▪ Compilador ejecutable Dalvik y corre en la Máquina Virtual Dalvik.</li> <li>▪ Soporta los siguientes formatos multimedia: WebM, H.263, H.264,3GP,MP4, MP3, MIDI, OggVorbis, WAV, JPEG, PNG, GIF y BMP.</li> </ul>

## 2.5.2 Apple iOS

**Tabla. 2.6. Sistema Operativo iOS.**

<b>Sistema Operativo</b>	<b>iOS</b>
<b>Fabricante</b>	Apple Inc. (2007)
<b>Versiones</b>	V.3.1.3.(iPhone Touch 1G) - V.4.2.1(iPhone 3G) - V.4.2.7(CDMA iPhone 4)- V.4.3.5 (iPhone 4, iPad, Apple TV )
<b>Compatibilidad con Dispositivos</b>	iPhone, iPhone Touch, iPhone 4, iPad, Apple TV
<b>Características Principales</b>	<ul style="list-style-type: none"> <li>▪ Multitarea (Multitasking): capaz de navegar por Internet, hacer algunos tweets, actualizar estado de Facebook al mismo tiempo.</li> <li>▪ Múltiples cuentas de Exchange (Multiple Exchange Accounts).</li> <li>▪ Cámara de 5x y zoom (5 X Camera Zoom).</li> <li>▪ Wi-Fi en modo de espera (Wi-Fi Stable In Sleep Mode): Con esta disposición, puedes recibir llamadas de Skype y otros servicios de VoIP en modo de suspensión.</li> <li>▪ iOS 4 SSL VPN: Capaz de asegurar el acceso VPN a través de tu navegador y otras aplicaciones basadas en la red.</li> <li>▪ Vídeo de enfoque automático (Video Autofocus): iPhone 4 incluye capacidad de grabación de vídeo HD 720p.</li> </ul>

### 2.5.3 BlackBerry OS

**Tabla. 2.7. Sistema Operativo BlackBerry.**

Sistema Operativo	BlackBerry
Fabricante	Research in Motion Company (2002)
Versiones	V.4.5 - V.4.6- V.5.0 - V.6.0- V.7.0
Compatibilidad con Dispositivos	Exclusivamente dispositivos BlackBerry
Características Principales	<ul style="list-style-type: none"> <li>▪ Proporciona multitarea y soporte para dispositivos de entrada especializados que se han adoptado por parte de RIM para su uso en dispositivos móviles.</li> <li>▪ Panel táctil y pantalla táctil.</li> <li>▪ Múltiples cuentas de Exchange (Multiple Exchange Accounts).</li> <li>▪ Soporte nativo para correo electrónico corporativo, a través de MIDP 1.0</li> <li>▪ Las actualizaciones del sistema operativo puede ser automáticamente disponibles en los operadores inalámbricos compatibles con el BlackBerry a través de la carga de software inalámbrico (OTASL) de servicio.</li> <li>▪ Los desarrolladores de terceros pueden escribir software usando la disposición BlackBerry API.</li> </ul>

### 2.5.4 Symbian OS

**Tabla. 2.8. Sistema Operativo Symbian.**

Sistema Operativo	Symbian
Fabricante	Symbian Ltd (2009)
Versiones	Symbian 1,2,3, Anna, Belle, 4
Compatibilidad con Dispositivos	Exclusivamente dispositivos Nokia
Características Principales	<ul style="list-style-type: none"> <li>▪ Cuenta con seis interfaces de usuario, opera en ROM y ha sido creado con el fin de ahorrar batería.</li> <li>▪ Se caracteriza por ser un sistema operativo extremadamente robusto y ahorrativo de recursos, tanto de la memoria como del procesador, permitiendo de esta manera aumentar la vida útil de la batería.</li> <li>▪ Permite la conectividad con diferentes dispositivos a través de Bluetooth, además ha mejorado la calidad de sus gráficos 3D.</li> <li>▪ Compatibilidad con otros sistemas de diversas cámaras digitales que posean hasta 5 megapíxeles.</li> <li>▪ Soportan aplicaciones Java MIDP 2.0, Symbian, C++, flash y Python</li> </ul>

### 2.5.5 Windows Mobile OS

**Tabla. 2.9. Sistema Operativo Windows Mobile.**

Sistema Operativo	Windows Mobile
Fabricante	Microsoft Corporation (2000)
Versiones	Pocket Pc 2000, 2002, Windows Mobile 2003, 5, 6, 6.1, 6.5, 7
Compatibilidad con Dispositivos	Pocket Pc (Palm Dell) , Windows Mobile (Nokia), HTC
Características Principales	<ul style="list-style-type: none"> <li>▪ Office Mobile, un conjunto de versiones móviles de aplicaciones de Microsoft Office, incluyendo Outlook Mobile.</li> <li>▪ Internet Explorer Mobile, un navegador de Internet desarrollado por Microsoft para Pocket PC y PC de mano que viene cargado por defecto con Windows Mobile y Windows CE para PC de mano.</li> <li>▪ Windows Media Player para Windows Mobile.</li> <li>▪ Cliente VPN para PPTP.</li> <li>▪ Conexión compartida a Internet (ICS), que en los teléfonos móviles permite que el teléfono para hacer su conexión a Internet a disposición de los ordenadores a través de USB y Bluetooth.</li> </ul>

## 2.6 COMPARACIÓN ENTRE SISTEMAS OPERATIVOS

### 2.6.1 Detalles básicos

Lo más importante de un sistema operativo es el núcleo (Kernel). Android usa un núcleo Kernel de Linux, con un entorno especial de Java. El iPhone basado en OS X, variante de Unix, que es uno de los sistemas operativos más compacto en el mundo. Symbian y Windows Mobile son sistemas operativos maduros y estables. Por último, BlackBerry OS usa un Kernel propio con un motor Java, muestra una mejora en la interfaz [12], sin embargo tiene algunas limitaciones sobre todo en su presentación y pantalla.

Debido a que ciertas versiones de los diferentes sistemas operativos aun están saliendo al mercado y no están posicionados, se ha hecho una comparativa con versiones ya probadas y establecidas de cada sistema operativo, visto en la Tabla 2.10.

Tabla. 2.10. Comparación de detalles básicos entre los SO de los Smartphones.

Sistema Operativo	Android 2.1 Eclair	BlackBerry OS 4.7	iPhone OS 3.0	Symbian 5th Edition	Windows Mobile 6.5
Apariencia visual					
Tipo de núcleo	Linux	Propietario	OS X	Symbian	Windows CE
Adaptabilidad	Excelente	Buena	Mala	Excelente	Excelente
Edad de la plataforma	Joven	Madura	Adolescente	Madura	Madura
Soporte para empresas	Joven	BlackBerry	Exchange	Exchange, Domino, BlackBerry	Exchange, Domino, BlackBerry
Tecnologías inalámbricas	GSM, WiFi	GSM, CDMA, WiFi	GSM, WiFi	GSM, WiFi	GSM, CDMA, WiFi

## 2.6.2 La interfaz de usuario

La interfaz de usuario y presentación es uno de los puntos que más se observa para decidir la compra de un Smartphone. En ese campo, el iPhone lleva una gran ventaja debido a su sistema operativo que viene trabajando hace muchos años en la calidad de video y apariencia. Con simples desplazamientos de los dedos, se puede realizar importantes y llamativos cambios entre aplicaciones. Android también permite usar gestos táctiles, pero los movimientos son difíciles de aprender porque son diferentes en cada aplicación, pero viene en mejora constante con nuevas y atractivas opciones gráficas [12].

Windows Mobile y BlackBerry OS son fáciles de usar: el primero gracias a su stylus, y el otro con las conocidas ruedas o bolitas de los móviles RIM, muy sencillas pero con pobre atractivo gráfico [12]. En Symbian, no se ha visto una mejora gráfica real para Smartphone y tienen todavía mucho que aprender de la competencia. Una de

las diferencias en la experiencia del interfaz se debe al tipo de pantalla del dispositivo, bien sea capacitiva o resistiva [12].

**Tabla. 2.11. Comparación de la interfaz de usuario entre los SO de los Smartphones.**

Sistema Operativo	Android 2.1 Eclair	BlackBerry OS 4.7	iPhone OS 3.0	Symbian 5th Edition	Windows Mobile 6.5
Gestos	Sí	Sí	Sí	Limitado	Limitado
Tecnología de la pantalla	Capacitiva	Capacitiva	Capacitiva	Resistiva / Capacitiva	Resistiva
Multitáctil	Sí	Sí	Sí	No	No
Cambios de temas	Sí	Sí	No	Sí	Sí
Obtención de información	Teclado virtual, teclado físico	Teclado virtual	Teclado virtual	Teclado virtual, T9, y triple clic; reconoce caracteres; teclado físico	Teclado virtual, reconoce caracteres, teclado físico

### 2.6.3 Funcionamiento

Una desventaja de Apple es la de no permitir la ejecución de más de una aplicación al mismo tiempo pero en su nueva versión para el iPhone 4 se están resolviendo estas dificultades. En el caso de Windows Mobile, la implementación de multitasking no ha sido de gran solución para resolver la multitarea, posee muchas dificultades de desempeño. Los demás sistemas operativos (Android, Symbian y BlackBerry) si permiten correr aplicaciones de fondo, con mayor o menor eficacia.

El servicio push es otro tema que interesa mucho a los usuarios. BlackBerry ha demostrado que su servicio de envío de mensajes es uno de los mejores del mercado. Windows Mobile no se queda atrás, y la integración con Exchange es muy sencilla y placentera para el consumidor. Android y Symbian incluyen envío de mensajes de una forma didáctica y atractiva, aunque no lucen por la accesibilidad ni facilidad de

implementación. iPhone OS 3.0, ofrece opciones que los usuarios demandaban y es la de MMS y Bluetooth estéreo [12].

**Tabla. 2.12. Comparación del funcionamiento entre los SO de los Smartphones.**

<b>Sistema Operativo</b>	<b>Android 2.1Eclair</b>	<b>BlackBerry OS 4.7</b>	<b>iPhone OS 3.0</b>	<b>Symbian 5th Edition</b>	<b>Windows Mobile 6.5</b>
<b>Notificación</b>	Pop-up, fondo	Pop-up, fondo	Pop-up	Pop-up	Bandeja, pop-up
<b>Administración de contactos</b>	Google,	BES, BIS	Exchange, ActiveSync, Mac OS	Exchange, Domino, BlackBerry,	Exchange, Domino, BlackBerry,
<b>Multitasking</b>	Sí	Sí	Si	Sí	Sí
<b>Soporte multimedia</b>	Android Music	iTunes sin DRM	iTunes	Ovi	Windows Media Player
<b>Búsqueda global</b>	No	No	Sí	Sí	No
<b>Actualización de firmware</b>	Tethered, OTA	Tethered, OTA	Tethered	Tethered, OTA	Tethered, OTA
<b>Motor del navegador</b>	WebKit	Propietario	WebKit	WebKit	Internet Explorer
<b>Bluetooth Estéreo</b>	Sí	Sí	Sí	Sí	Sí

#### 2.6.4 Desarrollo de terceros

El ganador en el año 2009 y medio año del 2010 indiscutiblemente es Apple. La tienda de iTunes es un éxito hoy en día, y funciona muy bien tanto como para pequeños desarrolladores como para grandes compañías. Ciertamente podrían mejorarla de varias maneras, pero al menos abrió la puerta para nuevos negocios. Sin embargo un gran competidor en cuanto a aplicaciones desarrolladas por terceros es Android, con su Android Market con más de 3 millones de aplicaciones descargadas [13]

Windows Mobile y Symbian posee una cantidad aceptable de aplicaciones, pero de una forma desordenada ya que no existe un lugar único para encontrarlas de manera fácil, cómoda y segura. Una parte fundamental en todos los sistemas operativos móviles es la tienda de aplicaciones. Mientras algunos como Android ya tienen su Market, otros

como BlackBerry recientemente están implementando de forma ordenada una tienda de aplicaciones [12].

**Tabla. 2.13. Comparación de desarrollo de terceros entre los SO de los Smartphones.**

<b>Sistema Operativo</b>	<b>Android 2.1 Eclair</b>	<b>BlackBerry OS 4.7</b>	<b>iPhone OS 3.0</b>	<b>Symbian 5th Edition</b>	<b>Windows Mobile 6.5</b>
<b>Disponibilidad de SDK / Soporte</b>	Sí	Sí	Sí	Sí	Sí
<b>Tienda de aplicaciones</b>	Sí	si	Sí	Próximamente	Sí
<b>Disponibilidad de aplicaciones</b>	Alta	Mediana	Alta	Mediana	Alta
<b>Aplicaciones nativas</b>	Si	No	Sí	Sí	Sí
<b>Administración local de aplicaciones</b>	Excelente	Buena	Excelente	Buena	Buena

No existe un teléfono que cubra todas las expectativas de los usuarios, y tampoco el sistema operativo que pueda aprovecharlo. Pero el desarrollo de los sistemas operativos jóvenes nos hace pensar que cada día tendremos un dispositivo móvil que más se acerque a nuestros requerimientos y nos permita personalizarlo a nuestros gustos.

Según las comparativas obtenidas de la investigación obtenemos que el sistema operativo Android, es una excelente plataforma para desarrollar aplicaciones, por ser software libre sin necesidad de adquirir licencias de programación o instalación evitándonos los altos costos por licencias y permisos, además porque se lo puede empezar a programar con tan solo un pequeño conocimiento de java bien sea en Linux o Windows en cualquier versión.

## **CAPITULO III**

### **SISTEMA OPERATIVO ANDROID**

#### **3.1 INTRODUCCIÓN**

El 5 de Noviembre del año 2007 se confirma la entrada del gigante Google en el mercado de la telefonía móvil [14], y lo hace de la mano con Open Handset Alliance es un grupo de 84 compañías de tecnología y móviles que se han unido para acelerar la innovación en los consumidores móviles y ofrecer una experiencia móvil más rica, más barato y mejor. Juntos han desarrollado Android, la primera plataforma móvil completa, abierta y libre. Los miembros incluyen a operadores móviles (Sprint, SoftBank, T-Mobile, Telecom, Telefónica, Vodafone, Telus), fabricantes de teléfonos (Acer, Alcatel, Dell, Foxconn, Fujitsu, HTC, Kyocera, Lenovo, LG, Motorola, Nec, Samsung, Sony Ericsson, Toshiba), empresas de semiconductores (ARM, Atheros, Intel, Nvidia, Qualcomm, Texas Instruments), empresas de software (Cooliris, eBay, Google, motoya, Nuance, Svox) y empresas de comercialización (Accenture, Teleca, Aplix, Borqs, Sasken) [15].

Una semana después de este anuncio aparecen los primeros SDK (Software Development Kits) para que cualquier persona que lo desee pueda realizar sus programas [14].

La estrategia de Google es la de dominar el mercado del software como su principal objetivo [14] y la de ir incursionando en el desarrollo de dispositivos como el Google Nexus One.

Android decide dirigir sus esfuerzos en el mundo de la telefonía móvil, optando por crear un sistema operativo, con aplicaciones incluidas apegadas a lo que tiene Google, y que mejor manera de realizar aplicaciones que disponer de un kit de desarrollo y de un concurso público para incentivar a los desarrolladores a migrarse a su plataforma.

Android es una oportunidad grande para escribir aplicaciones innovadoras para los dispositivos móviles. Este sistema operativo es una pila de software de software libre incluyendo aplicaciones dedicadas, junto con un grupo de librerías API para escribir aplicaciones móviles que pueden dar forma a la apariencia, sensación y la función de los teléfonos móviles [16].

En los últimos años, el mayor avance en el desarrollo del sistema operativo del teléfono móvil ha sido la introducción de MIDlets<sup>9</sup> Javahosted. Los MIDlets se ejecutan en una máquina virtual de Java, permitiendo a los desarrolladores crear aplicaciones que se ejecutan en la amplia variedad de hardware que soporta el tiempo de ejecución Java.

La introducción de MIDlets de Java ha ampliado el número de desarrolladores, pero la falta de acceso a hardware de bajo nivel y la seguridad de ejecución han hecho que la mayoría de las aplicaciones móviles sean programas de escritorio diseñados para funcionar en una pantalla más pequeña en vez de tomar ventaja de la movilidad inherente de la computadora de mano.

Android es uno de los sistemas operativos móviles diseñados para el hardware móvil cada vez más potentes. Windows Mobile y iPhone de Apple ofrecen ahora el desarrollo de aplicaciones móviles cada vez más amigables. Sin embargo, Android, prioriza las aplicaciones nativas en los creados por terceros y restringe la comunicación entre aplicaciones y datos nativos del teléfono. Android ofrece nuevas posibilidades de aplicaciones móviles al ofrecer un entorno de desarrollo abierto basado en una fuente abierta del Kernel de Linux. El acceso de hardware está disponible para todas las aplicaciones a través de una serie de bibliotecas de la API, y la interacción de aplicaciones, que cuidadosamente controlado, es totalmente compatible [16].

---

<sup>9</sup>Midlet es un programa en lenguaje de programación Java para dispositivos embebidos (se dedican a una sola actividad), más específicamente para la máquina virtual Java MicroEdition.

## 3.2 QUÉ ES ANDROID

El término "Android" tiene su origen en la palabra griega ANDR, que significa "hombre o mujer" y el sufijo EIDE, que se significa "por igual o de la especie". Esto junto significa "ser humano" [17].

Android es una pila de software para dispositivos móviles que incluye un sistema operativo, aplicaciones esenciales y middleware [18], lo que significa una referencia a un conjunto de programas del sistema o un conjunto de programas de aplicación que forman un sistema completo. Esta plataforma de software proporciona una base para las aplicaciones como una verdadera plataforma de trabajo.

El logotipo de Android fue diseñado con la fuente Droid, hecha por Ascender Corporation [19]. El verde es el color del robot Android que representa el sistema operativo. El color print es PMS 376C y color GBN en hexadecimal es #A4C639, como se especifica en la Android Brand Guidelines<sup>10</sup>.



**Figura. 3.1. Android SO en Teléfonos Móviles Inteligentes.**

## 3.3 ARQUITECTURA

La pila de software se divide en cuatro capas diferentes, que incluyen cinco grupos diferentes como se muestra en la Figura 3.2:

---

<sup>10</sup><http://www.android.com/branding.html>

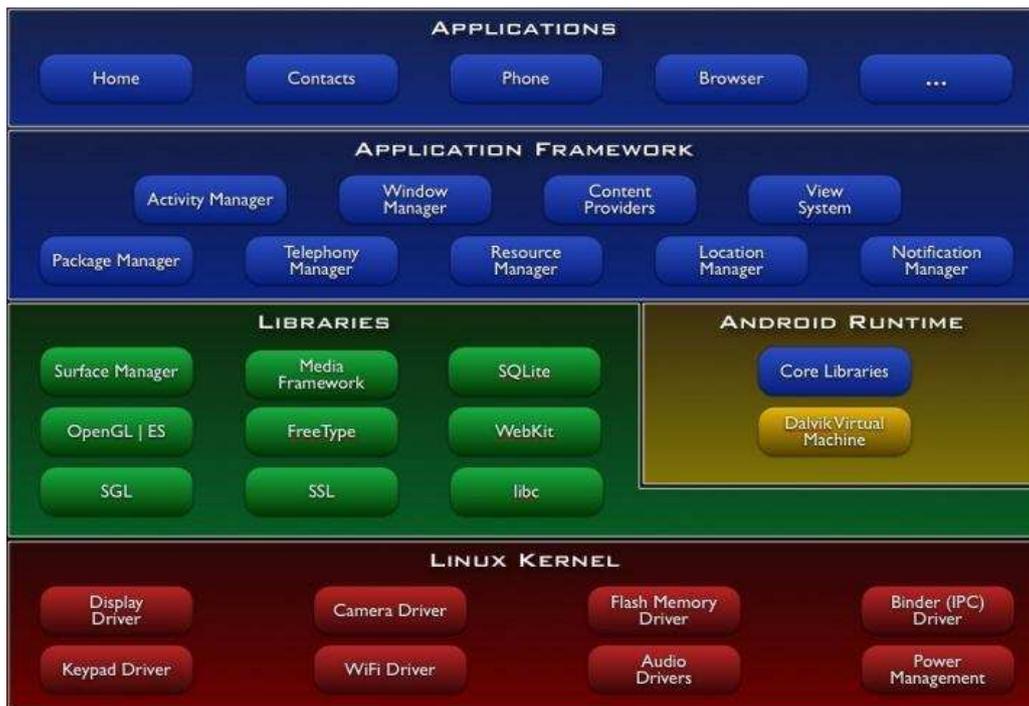


Figura. 3.2. Arquitectura del Sistema Operativo Android<sup>11</sup>.

### 3.3.1 Applications

La plataforma de software Android viene con un conjunto de aplicaciones básicas como navegador, cliente de correo electrónico, programa de SMS, mapas, calendario, contactos y mucho más. Todas estas aplicaciones se escriben usando el lenguaje de programación Java. Las aplicaciones se pueden ejecutar al mismo tiempo, es posible escuchar música y leer un correo electrónico al mismo tiempo. Esta capa en su mayoría es utilizada comúnmente por los usuarios de teléfonos celulares [20].

La capa más alta en el diagrama de la arquitectura de Android es la capa de aplicaciones y reproductores. Los usuarios finales sólo ven estos programas, sin ver toda la acción que sucede debajo de la capa de aplicaciones.

Las aplicaciones son programas que pueden hacerse cargo de toda la pantalla e interactuar con el usuario. Por otra parte, los Widgets, funcionan sólo en un pequeño rectángulo de la pantalla de la aplicación.

<sup>11</sup> <http://developer.android.com/guide/basics/what-is-android.html>

Cuando se adquiere un teléfono Android, llega con los programas por default de cada versión del sistema operativo, con un número de aplicaciones estándar del sistema, incluyendo las siguientes [20]:

- Teléfono de marcación.
- Correo electrónico.
- Contactos.
- Navegador Web.
- Android Market.

Usando el Android Market, el usuario será capaz de descargar nuevos programas para ejecutar en su teléfono.

### 3.3.2 Application Framework

Una aplicación Framework es un software que se utiliza para aplicar una norma estructurada de una aplicación para un sistema operativo específico. Con la ayuda de los administradores, proveedores de contenido y programadores de otros servicios que pueden volver a montar funciones utilizadas por otras aplicaciones existentes [20].

Encima de las librerías nativas y tiempo de ejecución, se encuentra la capa de aplicación Framework. Esta capa proporciona los bloques de construcción de alto nivel que va a utilizar para crear sus aplicaciones. El Framework viene pre instalado con Android, pero también se puede ampliar con sus propios componentes según sea necesario [20]. Las partes más importantes del Framework son los siguientes:

- **Activity Manager:** Controla el ciclo de vida de aplicaciones y mantiene un común "backstack" para la navegación del usuario.
- **Content Providers:** Estos objetos encapsulan datos que necesitan ser compartidos entre las aplicaciones, como los contactos.
- **Resource Manager:** Los recursos son cualquier cosa que vaya con su programa que no es el código.

- **Manager Location:** Un completo identificador de ubicación del teléfono Android.
- **Notification Manager:** Eventos como los mensajes que llegan, citas, alertas de proximidad y más se pueden presentar de una manera discreta para el usuario.

### 3.3.3 Libraries

Las bibliotecas están disponibles en todos los escritos en C / C++. Ellos serán llamados a través de una interfaz Java. Estos incluyen el Administrador de la superficie (para la composición de ventanas), gráficos 2D y 3D, los medios de comunicación códecs como MPEG-4 y MP3, la base de datos SQL SQLite y Web Kit navegador web motor [20]. Las librerías nativas más importantes son las siguientes:

- **Gestor de Superficie:** Android utiliza un gestor de ventanas de composición similar a Vista o Compiz, pero es mucho más simple. En lugar de dibujar informa directamente al búfer de pantalla; los comandos de dibujo entran en mapas de bits fuera de la pantalla que se combinan con mapas de bits para formar la pantalla que ve el usuario. Esto permite que el sistema cree todo tipo de efectos interesantes, como ver a través de ventanas y transiciones [20].
- **Gráficos 2D y 3D:** Elementos de dos y tres dimensiones se pueden combinar en una única interfaz de usuario con Android. La biblioteca utiliza hardware 3D si el dispositivo no tiene software de render rápido [20].
- **Códecs multimedia:** Android puede reproducir vídeo, grabar y reproducir audio en una variedad de formatos, incluyendo AAC, AVC (H.264), H.263, MP3, y MPEG-4[20].
- **Base de datos SQL:** Android incluye el ligero motor de base de datos SQLite 2, la misma base de datos utilizado en Firefox y en el Apple iPhone [20].
- **El motor del Navegador:** Para la visualización rápida de contenido HTML, Android utiliza el Web Kit library. Este es el mismo motor utilizado en el navegador Google Chrome, el navegador Safari de Apple, el iPhone de Apple, y la plataforma Symbian de Nokia[20].

Estas librerías no son aplicaciones que se destacan por sí mismos. Ellos existen sólo para ser llamados por los programas de nivel superior. A partir de Android 1.5, se puede escribir y desplegar sus propias bibliotecas nativas con el Kit Nativo de herramientas de desarrollo (NDK) [20].

### 3.3.4 Android Runtime

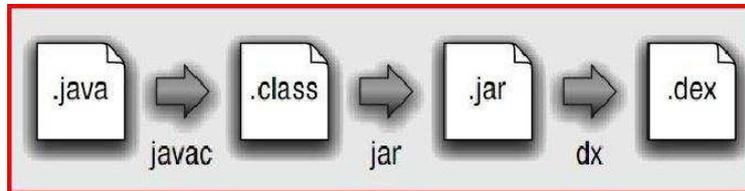
El tiempo de ejecución Android consta de dos componentes. En primer lugar un conjunto de bibliotecas del núcleo que proporciona la mayor parte de la funcionalidad disponible en las bibliotecas del núcleo del lenguaje de programación Java. En segundo lugar la máquina virtual Dalvik que funciona como un traductor entre el lado de la aplicación y el sistema operativo. Cada aplicación que se ejecuta en Android está escrita en Java. Como el sistema operativo no es capaz de entender este lenguaje de programación directamente los programas Java se traducen por la máquina virtual Dalvik [20].

La ventaja es que los diferentes programas no se afectan entre sí, así que un error de programa por ejemplo, puede conducir a un bloqueo del programa, pero no de todo el sistema.

También en la parte superior del núcleo esta el tiempo de ejecución de Android, incluyendo la máquina virtual Dalvik y el núcleo de las bibliotecas Java. La máquina virtual Dalvik es la aplicación de Google de Java, optimizado para dispositivos móviles. Todo el código escrito para Android será escrito en Java y se ejecutan dentro de la máquina virtual [20]. Dalvik difiere de Java en dos aspectos importantes:

- La máquina virtual Dalvik ejecuta los archivos (.dex), que se convierten en tiempo de compilación de la Clase (.jar) como se puede ver en la Figura 3.3. Los Archivos (.dex) son más compactos y eficientes que los archivos de clase, una consideración importante para las limitaciones de memoria y dispositivos a batería [20].

- El núcleo de las bibliotecas Java que viene con Android son diferentes tanto desde el Java Standard Edition (Java SE) y bibliotecas de Java Mobile Edition (Java ME) sin embargo hay una cantidad considerable de solapamiento [20].



**Figura. 3.3. Conversión del archivo .java a .dex.**

### 3.3.5 Linux Kernel

El Kernel de Linux es utilizado por Android para sus controladores de dispositivos, gestión de memoria, proceso de gestión y trabajo en red.

Android está construido sobre una base sólida y probada: el Kernel de Linux. Creado por Linus Torvalds en 1991, Linux proporciona la capa de abstracción de hardware para Android para ser transferido a una amplia variedad de plataformas en el futuro [20].

Internamente, Android utiliza Linux para su gestión de memoria, gestión de procesos, redes y otros servicios del sistema operativo. El usuario del teléfono Android nunca ve a Linux, y sus programas no harán que llame a Linux directamente; como desarrollador sin embargo, tendrá que tener en cuenta que está ahí.

Algunas utilidades necesarias durante el desarrollo interactúan con Linux. Los servicios básicos (incluidos los controladores de hardware, procesos, gestión de memoria, seguridad, red y administración de energía) son manejados por un Kernel de Linux. El núcleo también proporciona una capa de abstracción entre el hardware y el resto de la pila [20].

### 3.4 CARACTERÍSTICAS IMPORTANTES INTEGRADAS EN ANDROID

Android ofrece muchas características que cubren muchas áreas, tales como el desarrollo de aplicaciones, Internet, medios de comunicación y conectividad. Algunos de los más importantes se presentan en la siguiente lista [21].

- Application Framework permite la reutilización y sustitución de componentes.
- Máquina Virtual Davik optimizada para dispositivos móviles.
- Navegador integrado basado en el motor WebKit de código abierto.
- Gráficos optimizados alimentado por una colección de gráficos personalizados en 2D, gráficos 3D basado en OpenGL ES 1.0 (aceleración de hardware opcional).
- SQLite para almacenamiento de datos estructurados.
- Medios de apoyo para el audio comunes, videos y formatos de imagen (MPEG4,H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- Telefonía GSM (dependiente del hardware).
- Bluetooth, EDGE, 3G y WiFi (dependiente del hardware).
- La cámara, GPS, brújula y acelerómetro (dependiente del hardware).
- En el entorno de desarrollo incluye un emulador de dispositivos, herramientas para la depuración, la memoria, perfiles de rendimiento, y un plug-in para el IDE de Eclipse.

### 3.5 VERSIONES DE LA PLATAFORMA ANDROID

Android hace parte de los sistemas operativos con interfaz natural de usuario. El desarrollo de aplicaciones para esta plataforma se realiza mediante SDK, y el lenguaje de programación Java. Su código fuente está disponible bajo diversas licencias de software libre y código abierto destacando la versión 2 de la licencia Apache.

Curiosamente Android toma el nombre de varios Postres. A continuación las versiones más importantes de Android.

### 3.5.1 Android 1.5 (Cupcake)



**Figura. 3.4. Logo Oficial de Android 1.5 (Cupcake).**

Debido a las grandes mejoras introducidas en la tercera actualización de Android, de Abril de 2009, el número de versión saltó directamente a la 1.5. Basado en el kernel Linux 2.6.27 [22], las novedades más interesantes de esta versión eran las siguientes:

- Rediseño completo de todos los elementos de la interfaz.
- Interfaz de Android vía Android Developers.
- Transiciones animadas entre ventanas.
- Mejoras en la velocidad de la cámara.
- Menor tiempo de búsqueda de los satélites GPS, gracias a la posibilidad de utilizar A-GPS.
- Mejoras en la velocidad del navegador web gracias a la inclusión de la última versión de Webkit.
- Intérprete JavaScript.
- Añadió la posibilidad de copiar y pegar texto y buscar texto dentro de una página web.
- Posibilidad de personalizar los Widgets<sup>12</sup> mostrados en la pantalla de inicio.
- Inclusión de teclado en pantalla, con soporte para orientación vertical, horizontal, funcionalidades de auto corrección y soporte de diccionarios del usuario.
- Añadió la posibilidad de grabar y reproducir vídeos.
- Soporte de Bluetooth Estéreo.

---

<sup>12</sup>Widget: es una abreviación de las palabras window y gadget. Widget es una mini-aplicación de ordenador que se presenta como una pequeña ventana o caja. En general, se usa para tener acceso rápido a programas o funciones usadas frecuentemente, como la calculadora, reloj, calendario, o bien para conectar con otras aplicaciones disponibles en Internet.

### 3.5.2 Android 1.6 (Donut)



**Figura. 3.5. Logo Oficial de Android 1.6 (Donut).**

Versión lanzada en Septiembre de 2009, está basada en el Kernel 2.6.29 de Linux. Se considera una actualización menor, pero aun así introduce algunas novedades bastante interesantes [23]:

- Quick Search Box, es una caja de búsqueda en la pantalla de inicio que permite buscar entre distintas fuentes (los contactos, el historial del navegador, Google), con autocompletado y capacidad de aprendizaje.
- Mejorada velocidad de la cámara.
- Posibilidad de conectarse a redes VPN, 802.1x.
- Nueva pantalla para controlar la batería, que permite comprobar qué aplicaciones y servicios son los que más consumen. Desde esta pantalla se puede también parar o desinstalar estas aplicaciones.
- Las aplicaciones de Android Market aparecen ahora ordenadas por categorías (Aplicaciones, Juegos y Descargas). Para cada categoría podemos consultar las últimas actualizaciones y las aplicaciones más populares. Además para cada aplicación se muestra ahora capturas de pantalla y reviews de otros usuarios.
- Nuevo motor de texto a voz.

### 3.5.3 Android 2.0 / 2.1 (Eclair)



Figura. 3.6. Logo Oficial de Android 2.0 / 2.1 (Eclair).

**Android 2.0**, lanzada en Noviembre del 2009, se continuó con la tradición de utilizar dulces de repostería como nombres de versión. Las novedades son [24]:

- Rediseño en la interfaz del navegador, contando ahora con soporte para distintas características de HTML5 (entre ellas la etiqueta vídeo), la posibilidad de hacer zoom con una doble pulsación y thumbnails de los marcadores.
- Soporte nativo de flash para la cámara (aparentemente los fabricantes de teléfonos que contaban con flash previamente tuvieron que desarrollar sus propios drivers para superar esta limitación).
- Zoom digital, modo scene, balance de blanco, efectos de color y modo macro.
- Mejoras en el teclado virtual.
- Soporte para nuevos tamaños y resoluciones de pantalla.
- Contactos rápidos.
- Bluetooth 2.1.
- Soporte nativo de Facebook.
- Mejoras en Google Maps, que pasaba a ser multitáctil y soportar capas.
- Soporte de Microsoft Exchange.
- Mejoras en el calendario.

En Diciembre del 2009 se publicó una pequeña revisión, Android 2.0.1, que mejoraba la duración de la batería y la estabilidad, multillamadas, el GPS, el Bluetooth, y la velocidad de disparo y auto focus de la cámara [25].

**Android 2.1** llegó a los móviles Android en Enero de 2010, también se considera una actualización menor [26]. Entre otras cosas incluye:

- Reconocimiento de voz, permitiendo dictar en lugar de escribir en cualquier campo de texto.
- Mejoras en el teclado virtual.
- Galería 3D, al estilo Cover Flow.
- Uso del gesto de “pellizcar” para hacer zoom en el navegador, la galería y en Google Maps.
- Nuevas aplicaciones de reloj/tiempo y noticias.
- Mejoras en Google Maps: sincronización de sitios favoritos, modo noche y auto completado de búsquedas.
- Mejoras en la duración de la batería.

#### 3.5.4 Android 2.2 (Froyo)



**Figura. 3.7. Logo Oficial de Android 2.2 (Froyo).**

**Android 2.2** es una versión de plataforma de importantes cambios, que incluye características de uso, características de desarrollo, cambios en el API [27], y correcciones de errores que se muestra a continuación.

- Actualizaciones automáticas para aplicaciones: Las aplicaciones recibirán actualizaciones automáticas, consiguiendo tener siempre la última versión del software.
- Soporte WiFi IEEE 802.11n.

- Soporte para Radio FM.
- Soporte Flash 10.1.
- Soporte de la API gráfica OpenGL 2.0.
- Posibilidad de asignar un color de LED en el TrackBall para diferentes eventos del terminal.
- Mejora un 450% en Rendimiento frente a Eclair.
- Tethering por USB y hotspot WiFi.
- Compilador JIT (Just in Time) que hace unas 4 o 5 veces más rápida la nueva versión de Android.
- Más de 20 novedades enfocadas a las empresas, como estadísticas mundiales y bolsa.
- Application Data Backup API además de mensajes, e-mail y contactos; se puede conservar datos de las aplicaciones y las puntuaciones al cambiar de equipo.
- Cloud-to-Device API: Se puede enviar un mensaje a los servers y estos lo reenvían al móvil de manera optimizada. De esta manera se puede lanzar aplicaciones gracias a los Android Intents y un botón en el navegador que pone "Send to Phone". Por ejemplo, se puede abrir un mapa o una web que se está viendo en un ordenador directamente al móvil.
- Tethering & Portable Hotspot: Permite compartir la conexión del teléfono con el portátil, haciendo del móvil un hotspot.
- Nuevo Browser: Soporte para el nuevo V8, Javascript performance x3, El navegador más rápido del mundo actualmente. En el futuro incorporará gracias a HTML5 integración con la cámara, el acelerómetro o el reconocimiento de voz
- Soporte para Flash 10.1 y Adobe Air.
- Nuevo Market: Apps2SD, Update All, Búsqueda en los datos de una aplicación.
- Feedback: informe de bugs para desarrolladores.
- Android Market accesible desde la computadora: Permite acceder como usuario y reconoce automáticamente el teléfono con Android SO; se puede descargar una aplicación directamente al teléfono mediante OTA, como también permite comprar música desde el Market del ordenador y bajarla directamente a la PC. Si ya se tiene música en la PC, con un pequeño programa se puede tener todas las canciones disponibles en Streaming instantáneamente en el teléfono.

### 3.5.5 Android 2.3 (Gingerbread)



**Figura. 3.8. Logo Oficial de Android 2.3 (Gingerbread).**

La plataforma Android 2.3 presenta muchas características nuevas e interesantes para usuarios y desarrolladores [28].

- IU refinamientos de sencillez y rapidez: La interfaz de usuario se refina en muchos aspectos, en todo el sistema, por lo que es más fácil de aprender, rápido de usar y más eficiente energéticamente. Posee un tema visual simplificado de colores sobre fondo negro, contraste de la barra de notificación, los menús y otras partes de la interfaz de usuario que aportan viveza.
- Teclado más suave: Rediseñado y optimizado para la introducción de texto más rápido y de edición. Las mismas teclas se forman de nuevo y cambian de posición para una mejor orientación, lo que facilita ver y presionar con precisión, incluso a altas velocidades.
- Multitouch: Una nueva clave de acordes permite al usuario introducir rápidamente los números y símbolos. A partir de ciertas teclas, los usuarios también pueden acceder a un menú de caracteres acentuados, números y símbolos rápidos.
- Poder mejorar la gestión: El sistema Android tiene un papel más activo en la gestión de aplicaciones que mantienen el dispositivo despierto por mucho tiempo mientras se ejecuta en segundo plano.

- Control sobre las aplicaciones: Un acceso directo al control de administración de aplicaciones aparece ahora en el menú de opciones en la pantalla de inicio, por lo que es mucho más fácil de controlar y gestionar la actividad de la aplicación.
- Llamadas por Internet: El usuario puede hacer llamadas de voz a través de Internet a otros usuarios que tienen cuentas SIP. El usuario puede añadir un número de llamadas por Internet (la dirección SIP) a cualquier contacto y puede iniciar una llamada de contacto rápido. Para utilizar llamadas por Internet, el usuario debe crear una cuenta en el proveedor de SIP de su elección (las cuentas SIP no se proporcionan como parte de la función de llamadas por Internet). Además, el soporte para SIP de la plataforma de Internet y funciones de llamada en dispositivos específicos está determinado por sus fabricantes y las compañías asociadas.

### 3.5.6 Android 3.x



**Figura. 3.9. Logos No Oficiales de las versiones Android 3.X.**

(a) Honeycomb.      (b) Ice Cream Sandwich.

Las versiones a partir de Android 3.X aún no están oficialmente lanzadas al mercado y sus características no se las conoce a ciencia cierta, pero según la página de Android se vienen muchas mejoras en sus nuevas versiones en cuanto a aplicaciones para tablets, VoIP, 3D HD, nuevas redes de cuarta generación, WIMAX, LTE, entre otras [29].

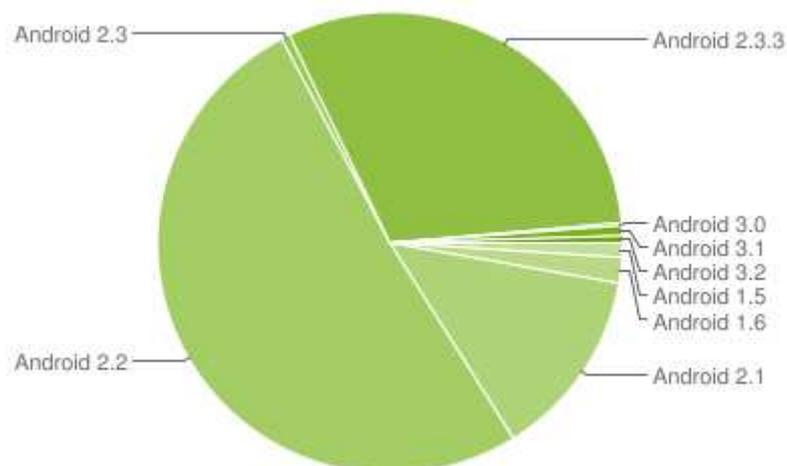
### 3.6 DISTRIBUCIONES DE LAS VERSIONES DE ANDROID.

#### 3.6.1 Distribución actual

La Tabla 3.1 y la Figura 3.10, se basa en el número de dispositivos Android que han accedido a Android Market en un plazo de 14 días hasta la fecha de recolección de datos se indican a continuación [30].

**Tabla. 3.1. Distribución de las Versiones de Android (Septiembre del 2011).**

Plataforma	Nivel API	Distribución
Android 1.5	3	1.0%
Android 1.6	4	1.8%
Android 2.1	7	13.3%
Android 2.2	8	51.2%
Android 2.3 -Android 2.3.2	9	0,6%
Android 2.3.3 -Android 2.3.4	10	30,7%
Android 3.0	11	0.2%
Android 3.1	12	0.7%
Android 3.1	13	0.5%



**Figura. 3.10. Distribución de las Versiones de Android (Septiembre del 2011).**

### 3.6.2 Android API Level

API Level es un valor entero que identifica únicamente el Framework API que ofrece una versión de la plataforma Android [31].

La plataforma Android proporciona un Framework API que las aplicaciones pueden ser utilizadas para interactuar con la base del sistema Android [31]. El Framework API se compone de:

- Una serie de paquetes básicos y las clases.
- Un conjunto de elementos y atributos XML para la declaración de un archivo de manifiesto.
- Un conjunto de elementos y atributos XML para declarar y acceso a los recursos.
- Un conjunto de intenciones.
- Un conjunto de permisos que pueden solicitar las aplicaciones, así como permisos de refuerzos incluidos en el sistema.

Cada versión sucesiva de la plataforma Android puede incluir cambios a la aplicación de Android Framework API que ofrece. Cambios al Framework API se han diseñado para que la nueva API siga siendo compatible con versiones anteriores de la API. Es decir, la mayoría de los cambios en la API son aditivos e introducen nuevas funcionalidades o reemplazos. Como parte de la API se actualizan, las partes más viejas son reemplazadas pero no se quitan, de modo que las aplicaciones existentes se pueden seguir utilizando. En un número muy pequeño de casos, las partes de la API pueden ser modificadas o eliminadas, aunque por lo general estos cambios sólo son necesarios para asegurar la robustez de la API y la aplicación o la seguridad del sistema. Todas las partes de la API de otras revisiones anteriores se llevan a cabo sin modificación [31].

El Framework API de referencia que ofrece la plataforma Android se especifica mediante un identificador entero llamado "API Level". Cada versión de la plataforma Android soporta exactamente un nivel de API, aunque se apoya implícitamente de todos los niveles anteriores de la API (hasta del nivel de API 1). La versión inicial de la

plataforma Android proporciona un nivel de API y versiones posteriores, se ha incrementado el nivel de API [31].

La Tabla 3.2 especifica el API level con el apoyo de cada versión de la plataforma Android.

**Tabla. 3.2. API Level.**

PlatformVersion	API Level	VERSION_CODE
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

## **CAPITULO IV**

### **PROGRAMACIÓN EN ANDROID**

#### **4.1 INTRODUCCIÓN**

La mayor ventaja en cuanto a la aparición del Sistema Operativo Android se la llevan los desarrolladores, ya que al no existir las restricciones impuestas por los sistemas operativos propietarios sobre el desarrollo de aplicaciones por terceros en Android, son libres de escribir aplicaciones que aprovechen al máximo el hardware de estos dispositivos para distribuirlas en un mercado abierto.

En parte por este creciente interés de los desarrolladores de software por los dispositivos con Android SO, las ventas de teléfonos han subido considerablemente teniéndose como principales fabricantes a HTC, Motorola, LG, Samsung y Sony Ericsson, a la vez que se suma recientemente al mercado de los dispositivos móviles el gigante creador de Android, Google, con su Google Nexus One el cual puede ser comprado en cualquier parte del mundo a través de su página web [32].

Android, al estar construido sobre un marco de código abierto y al disponer de potentes librerías dentro de su SDK permite a los desarrolladores experimentados expandir la plataforma, aprovechando las características únicas para mejorar los productos existentes o crear otros nuevos e innovadores.

En este capítulo se explicará los temas más relevantes usados para la programación de la aplicación de reconocimiento utilizando el SDK de Android proporcionado por Google, herramientas como el Eclipse IDE y el lenguaje de programación Java para crear aplicaciones sencillas que puedan introducir al lector en el mundo del desarrollo de aplicaciones para Android.

Todo lo que se necesita para empezar a escribir aplicaciones para Android es una copia del SDK de Android y el kit de desarrollo Java y el Eclipse IDE el cual es recomendado por Android, sin embargo se puede hacer en cualquier otro IDE de Java.

Para explorar Android desde la comodidad de cualquier sistema operativo, el SDK incluye un emulador para los tres entornos de sistemas operativos, ya que las aplicaciones de Android se ejecutan en un dispositivo virtual o AVD, no hay ninguna ventaja ni desventaja al desarrollar en un sistema operativo en particular. El código de Android está escrito con la sintaxis de Java, y las bibliotecas del núcleo de Android incluyen la mayoría de las API básicas de Java.

Las aplicaciones antes de que se puedan ejecutar, son traducidas en código de bytes Dalvik [33]. Como resultado, se obtiene el beneficio de la utilización de Java, mientras que sus aplicaciones tienen la ventaja de correr en una máquina virtual optimizada para dispositivos Android.

La descarga del SDK incluye todas las bibliotecas de Android, la documentación completa, y excelentes aplicaciones de muestra. También incluye herramientas para ayudar a escribir y depurar las aplicaciones (El emulador de Android para ejecutar sus proyectos utiliza la depuración Dalvik Servicio de Monitoreo – DDMS)[33].

Se aclara que por su pequeño factor de forma, hardware limitado, y memoria restringida de un teléfono móvil, se crean algunos problemas al diseñar aplicaciones. Se debe considerar que algunas de las cosas que se pueden realizar en una PC de escritorio o laptop, no se lo va a poder realizar en un móvil por su limitación de memoria y procesamiento.

Para el desarrollo en Android, el SDK incluye todas las herramientas y API's que necesita para escribir aplicaciones móviles convincentes y poderosas. El desafío más grande con Android, como sucede con cualquier conjunto de herramientas de desarrollo nuevo, es el aprendizaje de características y limitaciones de su API [33].

Si se tiene experiencia en el desarrollo con Java, se encontrará que las técnicas, la sintaxis y la gramática que se ha estado utilizando se traducirán directamente en Android, aunque algunas de las técnicas de optimización pueden parecer contradictorias. Si no se tiene experiencia con Java, pero si se ha utilizado otros lenguajes orientados a objetos como C++, la transición será sencilla. El poder de Android viene de su API, mas no de Java.

## 4.2 PREPARACIÓN DEL ENTORNO DE DESARROLLO PARA ANDROID

### 4.2.1 Requerimientos previos

Como las aplicaciones de Android se ejecutan en la Máquina Virtual Dalvik, se puede escribir sobre cualquier plataforma que soporte las herramientas de desarrollo [33]. Esto incluye actualmente los siguientes sistemas operativos vistos en la Tabla 4.1.

**Tabla. 4.1. Plataformas que soportan Herramientas de Desarrollo para Android.**

Android	Versiones
Microsoft Windows OS	XP, Vista o 7
Mac OS	10.4.8 o posterior(sólo los chips de Intel)
Linux	Toda versión

Para empezar, se tendrá que descargar e instalar lo siguiente:

- Eclipse IDE (Sección 4.2.2).
- El SDK de Android (Sección 4.2.3).
- Java Development Kit (JDK)<sup>13</sup> 5 o 6.

### 4.2.2 Eclipse IDE

Al utilizar Eclipse IDE para el desarrollo de Android ofrece algunas ventajas significativas. Eclipse es un IDE(entorno de desarrollo integrado) de código abierto particularmente popular para el desarrollo Java [33]. Está disponible su descarga, para

<sup>13</sup><http://java.sun.com/javase/downloads/index.jsp>

cada una de las plataformas de desarrollo con el apoyo de Android (Windows, Mac OS y Linux) de la página web de la Fundación Eclipse<sup>14</sup>.

Hay muchas variaciones disponibles cuando se selecciona la descarga de Eclipse, las siguientes son las configuraciones recomendadas para Android:

- Eclipse 3.3, 3.4 (Ganímedes).
- Eclipse JDT plug-in.
- WST.

WST y el JDT plug-in se incluyen en la mayoría de paquetes de Eclipse IDE.

### **Instalación de Eclipse**

La instalación de Eclipse consiste en:

- Descomprimir la descarga en una carpeta nueva.
- Ejecutar el archivo (.exe) de Eclipse.
- Cuando se inicia por primera vez, se crea un nuevo espacio de trabajo para el desarrollo de Android.

#### **4.2.3 SDK de Android**

El SDK de Android es completamente abierto. No hay costo alguno para descargar y utilizar la API, Google no cobra por permitir la distribución de sus programas finalizados. Se puede descargar la última versión del SDK para la plataforma de desarrollo de la página principal<sup>15</sup>.

El SDK se presenta como un archivo .ZIP que contiene las bibliotecas de la API, herramientas de desarrollo, documentación y varias aplicaciones. Se descomprime

---

<sup>14</sup>[www.eclipse.org/downloads/](http://www.eclipse.org/downloads/)

<sup>15</sup><http://code.google.com/android/download.html>

e instala el SDK en una carpeta nueva, se toma nota de su ubicación, ya que se necesitará más adelante [33].

El SDK contiene tres directorios principales. El primero es `.docs`, que tiene toda la documentación necesaria para empezar a desarrollar con la plataforma desde cero. Se empieza con el “Getting started” para empezar a programar.

A continuación se tiene “Desarrollando Aplicaciones“, donde se explica cómo implementar una interfaz gráfica, acceso a datos, y otros recursos. Por último, el “*toolbox*” o caja de herramientas para desarrolladores donde se explica sobre la filosofía de diseño, construcción de componentes, API’s opcionales y servicios así también como las API’s de Google. Además incluye información de referencia muy completa, con la lista de paquetes (muchos conocidos de Java), índice y jerarquía de clases [33].

Los ejemplos y las instrucciones paso a paso en la página de Android, siempre están dirigidos a utilizar Eclipse IDE como la herramienta de desarrollo de Android, con su plug-in(ADT). A pesar de ello se puede usar cualquier editor de texto o IDE Java como: NetBeans, DrJava Java IDE, BlueJ Java IDE, JIPE Java IDE, JCreator Java IDE LE (Light Edition) entre otros.

#### **4.2.4 El ADT**

El ADT (Android Developer Tool) para Eclipse, simplifica el desarrollo de Android mediante la integración de herramientas como el emulador y conversor de clase a (.dex), directamente desde el IDE. El ADT, crea, prueba y depurar sus aplicaciones más rápido y más fácil. El ADT tiene integrado en Eclipse lo siguiente [33]:

- Un Android Project Wizard, el cual simplifica la creación de nuevos proyectos e incluye una plantilla de aplicación básica.
- Editores basada en formularios de manifiesto, la disposición y recursos para ayudar a crear, editar y validar sus recursos XML.

- Estructura automatizada de los proyectos de Android, la conversión a archivos ejecutables Android (.dex), empaquetadores de archivos (.apk), y la instalación de paquetes en la máquina virtual Dalvik.
- El emulador de Android, incluye el control de la apariencia del emulador, la configuración de conexión de red, y la capacidad de simular llamadas entrantes y mensajes SMS.
- El Servicio de Monitoreo y Depuración Dalvik (DDMS), incluye el reenvío de puertos, pila, visualización de hilos, detalles del proceso, y la captura de pantalla de las instalaciones.
- Acceso al dispositivo o sistema de archivos del emulador, lo que le permite navegar por el árbol de carpetas y archivos de transferencia.
- Depuración en tiempo de ejecución, permitiendo establecer puntos de interrupción y ver las pilas de llamadas.
- Todos los registros Android / Dalvik y salidas de la consola se muestran en la Figura 4.1, además la perspectiva DDMS dentro de Eclipse con el ADT plug-in instalado.

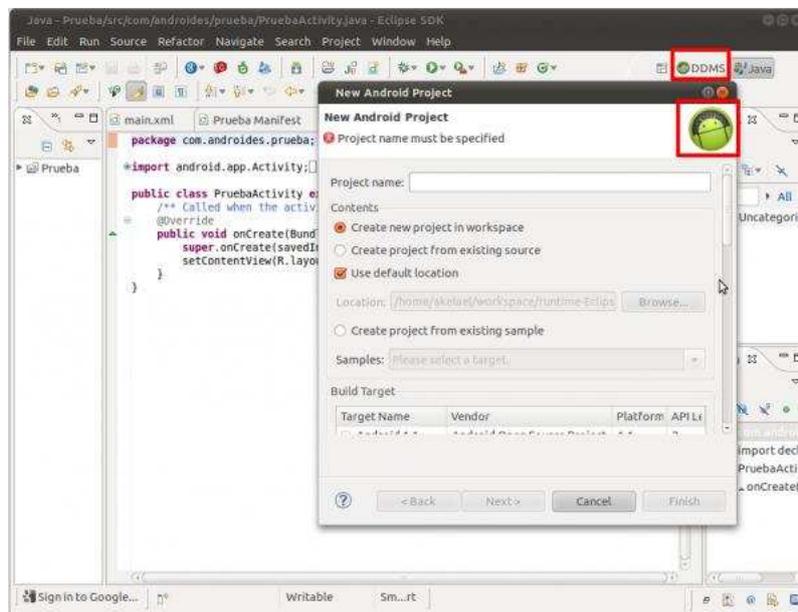
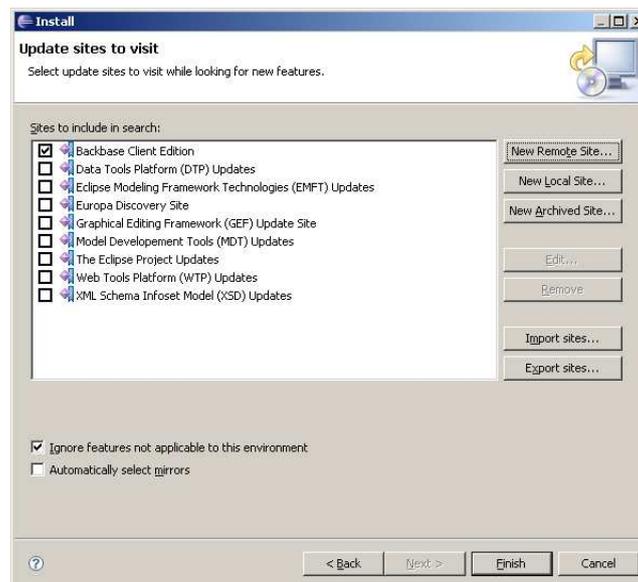


Figura. 4.1. Perspectiva DDMS dentro de Eclipse con el ADT plug-in instalado.

## Instalación del ADT Plug-in

Para instalar las herramientas de desarrollo de ADT plug-in se sigue los siguientes pasos [33]:

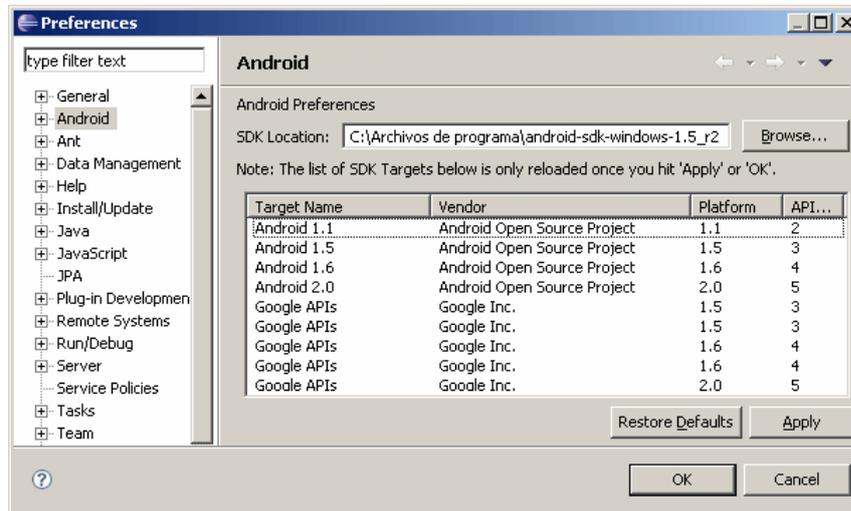
1. Seleccionar: Ayuda ⇨ Actualizaciones de software ⇨ Buscar e instalar (dentro de Eclipse).
2. En el cuadro de diálogo resultante, se selecciona: Buscar características nuevas para instalar.
3. Seleccionar: Nuevo sitio remoto, y entrar en la siguiente dirección en el cuadro de diálogo, como se muestra en la Figura 4.2:



**Figura. 4.2. Ventana configuración de sitio de actualización.**

4. Eclipse automáticamente descargará el plug-in. Cuando finaliza, se selecciona Android plug-in y se despliega las herramientas de desarrollo del cuadro de diálogo de búsqueda de los resultados y se hace clic en siguiente.
5. Se lee y acepta los términos del contrato de licencia y se hace clic en siguiente y luego en finalizar.
6. Cuando se haya terminado, se tendrá que reiniciar Eclipse y actualizar las preferencias de ADT. Luego se Reinicia y se selecciona Ventana ⇨ Preferencias (Eclipse o Preferencias de Mac OS).

7. A continuación, se selecciona Android en el panel izquierdo.
8. Se hace clic en Examinar: Se navegar a la carpeta en la que ha descomprimido el SDK de Android y se hace clic en Aplicar y Aceptar, como se muestra en la Figura 4.3.



**Figura. 4.3. Ventana de Preferencias.**

Si se descarga una nueva versión del SDK y se coloca en un lugar diferente, se tendrá que actualizar esta preferencia del SDK con el que el ADT se debe construir.

## 4.3 INTRODUCCIÓN AL DESARROLLO PARA ANDROID

Antes de empezar a escribir aplicaciones para Android, es importante entender cómo están construidas y su comprensión del ciclo de vida de aplicaciones Android. A continuación se verá cómo y por qué se debe utilizar los recursos externos, antes de una introducción al componente de la actividad.

### 4.3.1 Tipos de aplicaciones de Android

Las aplicaciones complejas son difíciles de encasillar en una sola categoría, por lo general incluyen elementos de cada uno de estos tipos. Al crear una aplicación se debe tener en cuenta, cómo es probable que la utilicen y luego diseñarla [33]. La mayoría de las aplicaciones que se creen en Android se ubicará en una de las siguientes categorías:

- **Foreground Applications:** Al crear aplicaciones de primer plano se necesita considerar cuidadosamente el ciclo de vida de actividad, para que la actividad pueda cambiar a la perfección entre el primer plano y el trasfondo.

Las solicitudes tienen poco control sobre sus ciclos de vida, y una aplicación de fondo, sin correr servicios es un candidato ideal para la limpieza de la gestión de recursos de Android. Esto significa que se necesita guardar el estado de la aplicación cuando ya no está en primer plano, para que se pueda presentar el mismo estado exacto cuando se la ponga al frente [33].

- **Background Services Intent Receivers:** Estas aplicaciones se ejecutan en segundo plano, con muy poca entrada de usuarios. A menudo se escuchan mensajes y la acción generada por el hardware, el sistema, o en otras aplicaciones [33].

Es posible crear servicios totalmente invisibles, pero en la práctica es mejor proporcionar al menos algún tipo de control de usuario. Como mínimo se debe permitir a los usuarios confirmar que el servicio está en funcionamiento y se debe dejar de configurar, poner en pausa, o darlo por terminado, según sea necesario.

- **Intermittent Applications:** A menudo, se creará una aplicación que reacciona a la entrada del usuario, pero sigue siendo útil cuando no está activa en el primer plano. Chat en vivo y aplicaciones de correo electrónico son ejemplos típicos. Estas aplicaciones son por lo general de un conjunto de Actividades y Servicios visible-invisible de fondo [33].

Dicha solicitud tiene que ser consciente de su estado cuando se interactúa con el usuario. Esto podría significar que la actualización de la interfaz de usuario de actividad es visible y el envío de notificaciones para mantener al usuario actualizado está en el fondo o invisible.

- **Widgets:** En algunos casos, su aplicación puede consistir enteramente de un componente flash. El uso de Widgets, puede crear componentes visuales interactivos que los usuarios pueden añadir a las pantallas de inicio. Las aplicaciones Widgets sólo se utilizan para mostrar información dinámica, como los niveles de batería, las previsiones meteorológicas o la fecha y hora [33].

### 4.3.2 Componentes de una aplicación Android

El componente de la actividad posiblemente es el más importante de los bloques de construcción para Android, la clase actividad, es la base de todas sus pantallas de interfaz de usuario [33]. Las aplicaciones para Android se componen de elementos de acoplamiento flexible, vinculado con un manifiesto de proyecto que se describe a continuación, cada componente y cómo interactúan.

Hay seis componentes que proporcionan los elementos necesarios para sus aplicaciones:

1. **Actividades:** Se encuentra en la capa de presentación de una aplicación. Cada pantalla de la aplicación será una extensión de la clase Actividad. Las Actividades usan vistas para formar interfaces de usuario gráficas que muestran la información y respuesta a las acciones del usuario [33]. En términos de desarrollo de escritorio, una actividad es equivalente a un formulario.
2. **Servicios:** Realizan la actualización de las fuentes de datos, actividades visibles y provocan notificaciones [33]. Se utilizan para realizar el proceso regular que debe continuar, aun cuando las actividades de su aplicación no sean activas o visibles.
3. **Proveedores de contenido:** Son un almacén de datos compartidos. Los proveedores de contenido se usan para gestionar y compartir bases de datos de aplicaciones. Son la forma preferida de intercambio de datos a través de los límites de aplicación [33]. Esto significa que se puede configurar sus propios proveedores de contenido para permitir el acceso de otras aplicaciones y el uso de los proveedores de contenidos expuestos por los demás para acceder a sus datos almacenados.

Los dispositivos Android incluyen varios proveedores de contenido nativo que exponen las bases de datos útiles como información de contacto.

4. **Intents:** Puede transmitir mensajes a todo el sistema o en una actividad de destino o servicio, indicando su intención de tener una acción realizada [33]. Entonces, el sistema determinará el destino que llevará a cabo cualquier acción en su caso.
5. **Receptores de radiodifusión:** Al crear y registrar un receptor de radiodifusión, su aplicación puede detectar Intenciones de difusión que responden a los criterios específicos de filtro [33]. Los Receptores de radiodifusión iniciarán automáticamente la aplicación para responder a una intención de entrada, lo que es ideal para el evento de aplicaciones.
6. **Notificaciones:** Permite la señal de los usuarios sin robar el enfoque o interrupción de las actividades actuales. Son las técnicas requeridas para conseguir la atención de los usuarios dentro de un servicio de receptor o difusión. Por ejemplo cuando el dispositivo recibe un mensaje de texto o llamada entrante, esta actividad las recibe por medio de la emisión de luces, haciendo sonidos, enviando iconos, o mostrando mensajes de dialogo. Se puede desencadenar este mismo hecho o evento desde tu propia aplicación usando notificaciones [33].

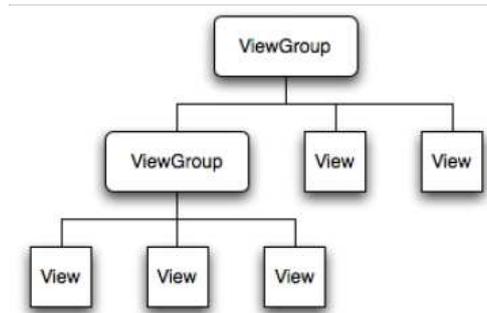
### 4.3.3 Interfaces de usuario para Android

Interfaces de usuario (UI) en Android se pueden construir en dos formas, mediante la definición de Código-XML o escrito en Código Java. La definición de la estructura GUI en XML es muy preferible, ya que la interfaz de usuario siempre debe ser separada de la lógica del programa. Además la adaptación de un programa de resolución de pantalla de uno a otro es mucho más fácil. La definición de una interfaz de usuario en XML es muy similar a crear un documento HTML común. Lo mismo que en Android XML-Diseños [33], [34]. Todo está bien estructurado y puede ser expresada por estructuras arbóreas:

## Jerarquía de los Elementos de la pantalla

La unidad básica funcional de una aplicación Android es la actividad, un objeto de la `android.app.Activity` class. Es una actividad que puede hacer muchas cosas, pero por sí mismo no tiene una presencia en la pantalla. Para dar a la actividad una presencia en la pantalla y el diseño de su interfaz de usuario, se trabaja con `View` y `Viewgroups` que son unidades básicas de la expresión de la interfaz de usuario en la plataforma Android.

- 1. View:** Es un objeto extender del `android.view`. Es una estructura de datos cuyas propiedades almacenan el diseño y el contenido de un área específica rectangular de la pantalla. Un objeto de vista se encarga de la medida, su diseño, dibujo, el desplazamiento y gestos para el área de la pantalla que representa. La clase `View` sirve como clase base para todos los `Widgets` (conjunto de subclases aplicado plenamente que atraen a elementos interactivos de la pantalla) [34]. Los `Widgets` manejan su propia medida y dibujo, así que se puede utilizar para construir su interfaz de usuario más rápidamente. La lista de `Widgets` disponibles incluye `TextView` es decir, `EditText`, `Button`, `RadioButton`, `CheckBox`, `ScrollView`.
- 2. Viewgroup:** Un `Viewgroup` es un objeto de clase `android.view.ViewGroup`. Como su nombre lo indica, un `Viewgroup` es un tipo especial de objeto de la vista, cuya función es contener y gestionar un conjunto de puntos de vista y subordinados. `Viewgroups` permite agregar a la estructura de la interfaz de usuario, la acumulación de elementos complejos de pantalla que puede ser abordado como una sola entidad. La clase `Viewgroup` sirve como clase base para los diseños de un conjunto de subclases aplicado plenamente que ofrecen tipos más comunes de diseño de la pantalla [34]. Los diseños ofrecen una manera de construir una estructura de una serie de puntos de vista.
- 3. A Tree Structured UI:** En la plataforma Android, se define la interfaz de usuario de una actividad con un árbol de vista y nodos `Viewgroup`, como se muestra en la Figura 4.4. El árbol puede ser tan simple o tan complejo, dependiendo que se necesite para hacerlo [34]. Se puede construir utilizando el conjunto de `Widgets` de Android o con diseños predefinidos.



**Figura. 4.4. Android UI – Estructura de Árbol.**

Para sujetar el árbol a la pantalla, para la representación, su actividad llama a su método `setContentView()` y pasa una referencia al objeto nodo raíz. Una vez que el sistema Android tiene la referencia al objeto nodo raíz, se puede trabajar directamente con el nodo de invalidar, medir y dibujar el árbol. Cuando su actividad se activa y recibe el foco, el sistema notifica a su actividad, y pide al nodo raíz medir y dibujar el árbol. El nodo raíz solicita que sus nodos secundarios se dibujen a su vez. Cada nodo ViewGroup en el árbol es responsable de la elaboración de sus hijos directos [34]. En la Tabla 4.2 se muestra algunas similitudes entre comandos de View Android y View Swing de Java.

**Tabla. 4.2. Comparación de comandos de ViewAndroid y View Swing.**

Android	Swing
Activities	JFrame
Views	JComponents
TextViews	JLabel
EditTexts	JTextFields
Buttons	JButtons

### 4.3.4 El archive `AndroidManifest.xml`

#### **Application Manifest**

Cada proyecto de Android incluye un archivo `AndroidManifest.xml`, que almacena en la raíz la jerarquía del proyecto. El Manifest permite definir la estructura e ingresar datos de las aplicaciones, y de los componentes [34].

Esto incluye nodos por cada uno de los componentes (Activities, Services, Content Providers, y Broadcast Receivers) que forman la aplicación, usando Intent Filters y Permissions, permite determinar la manera cómo interactúan entre ellos y con otras aplicaciones. Esto también ofrece a los atributos ingresar datos de aplicaciones (como el icono o tema) y adicionalmente top-level nodes puede ser usado para la configuración de seguridad y como pruebas de unidad, como se describe a continuación.

El Manifest incluye nodos que definen los componentes de la aplicación, configuración de seguridad y tipos de pruebas que forman la aplicación. Estos nodos son [33]:

- 1. Application:** El Manifest contiene un solo nodo de aplicación. Estos usan atributos para especificar los datos de su aplicación (incluyendo títulos, iconos, y tema). Este también actúa como contenedor que incluye Activity, Service, Content Provider, y Broadcast Receiver tags se usa para especificar los componentes de la aplicación [33].
- 2. Uses-permission:** Como parte del modelo de seguridad, uses-permission tags, indica los permisos determinados que la aplicación necesita para operar apropiadamente. Los permisos que incluyen serán presentados al usuario, para otorgar o negar, durante la instalación. Los permisos son necesarios para muchos de los servicios nativos de Android, particularmente aquellos con una implicación de costos o de seguridad (tales como la marcación, la recepción de SMS, o utilizar los servicios basados en ubicación) [33].

- 3. Permission:** Antes de que se pueda restringir el acceso a un componente de aplicación, es necesario definir un permiso en el Manifest. Se usa el Permission tag para crear el permiso. Los componentes de la aplicación se pueden requerir al añadir android:permissionattribute. Otras aplicaciones tendrán que incluir una etiqueta de uso de permiso en sus manifiestos antes de que puedan utilizar estos componentes protegidos. Dentro de Permission se puede especificar el nivel de acceso a la autorización (normal, dangerous, signature, signatureOrSystem), una etiqueta, y un recurso externo que contiene la descripción que explica los riesgos de la concesión de este permiso [33].
  
- 4. Instrumentation:** Proporcionan un marco para la realización de pruebas sobre las actividades y servicios en tiempo de ejecución. Se proporcionan enlaces para supervisar la aplicación y la interacción con los recursos del sistema [33]. Se crea un nuevo nodo para cada una de las clases de prueba que se haya creado para la aplicación.

### Manifest Editor

El ADT plug-in incluye un editor de Manifest visual para gestionar su manifiesto, en lugar de tener que manipular el código XML subyacente directamente. Para usar el Editor de manifiesto en Eclipse, se hace clic en el archivo AndroidManifest.xml, en su carpeta de proyecto, y se selecciona abrir con el Editor Manifest Android [33]. Esto presenta la pantalla de resumen de Manifest de Android, como se muestra en la Figura 4.5. Da una vista de alto nivel de la estructura de la aplicación y proporciona enlaces de acceso directo a las aplicaciones, permisos, instrumentación y pantallas XML [34].



Figura. 4.5. Android Manifest Overview.

En la Tabla de aplicaciones, que se muestra en la Figura 4.6. Se puede administrar el nodo de aplicación y la jerarquía de aplicación de componentes. Se puede especificar los atributos de una aplicación incluyendo su icono, la etiqueta, y el tema en el panel de la aplicación de atributos. El árbol de la aplicación permite administrar los componentes de aplicación, incluyendo sus atributos y los nodos secundarios asociados a la intención del filtro.

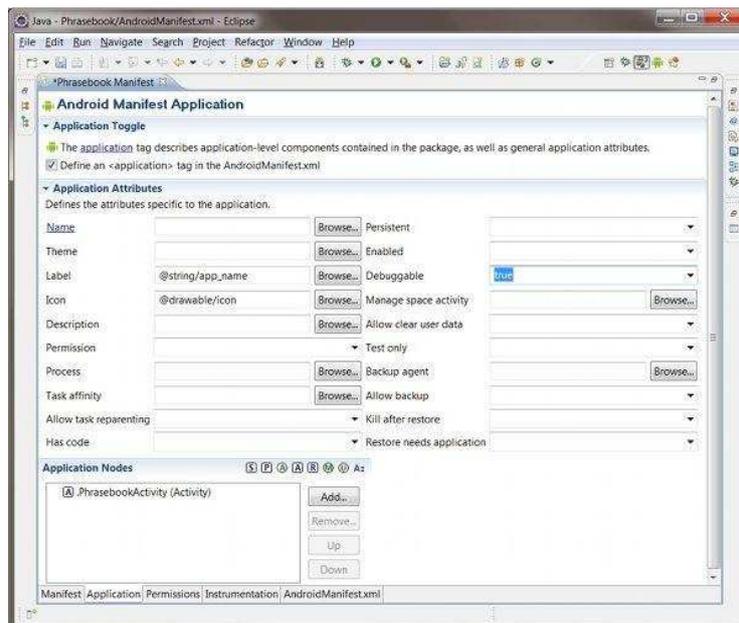


Figura. 4.6. Android Manifest Application.

### 4.3.5 Recursos de la Aplicación

Los recursos son archivos externos (archivos no código) que son usados por el código y compilados en un mismo tiempo dentro de la aplicación. Android soporta un número de diferentes clases de archivos de recursos, incluyendo XML, PNG, y archivos JPEG. El archivo XML tiene muchos formatos diferentes dependiendo en qué se escriba. Los recursos son externalizados de su código fuente, y el archivo XML es compilado en binario con formato de carga rápida por razones de eficiencia [34]. Los Strings son comprimidos dentro de sí mismos para un formato de almacenamiento más eficiente.

#### Creación de Recursos

Los Recursos de las aplicaciones se almacenan en res/folder de la jerarquía del proyecto. En esta carpeta, cada uno de los tipos de recursos disponibles puede tener una subcarpeta que contiene sus recursos.

Si se comienza un proyecto mediante el asistente de ADT, se creará una carpeta que contiene subcarpetas res de values, drawable-ldpi, drawable-mdpi, drawable-hdpi, y layoutresources, que contiene el diseño predeterminado, icono de la aplicación, y la cadena de recursos respectivamente [33], como se muestra en la Figura 4.7.



**Figura. 4.7. Carpetas y subcarpetas de Recursos de las Aplicación.**

Hay siete tipos de recursos primarios que tienen diferentes carpetas: simple values, drawables, layouts, animations, XML, styles y rawresources. Cuando la aplicación está desarrollada, estos recursos se compilan como la eficiencia más eficiente posible que se incluyen en el paquete de aplicación [33].

Este proceso también crea un archivo de clase R que contiene referencias a cada uno de los recursos que se incluye en el proyecto explicado en la sección 4.4.2. De este modo, se hace referencia a los recursos de su código, con la ventaja de comprobación de sintaxis en tiempo de diseño.

En todos los casos, los archivos de recursos deben contener sólo letras en minúscula, números, el punto y guión bajo [33].

### Creando Simple Values

Los valores admitidos son simples cadenas, colores, dimensiones, y una cadena o arrays de enteros. Todos los valores simples son almacenados en archivos XML en la resolución de carpeta values. Dentro de cada archivo XML, se indica el tipo de valor que se almacena y el uso de etiquetas. Por convención, los recursos están separados en distintos archivos para cada tipo.

### Strings

La externalización de las cadenas ayuda a mantener la coherencia dentro de su aplicación y la hace mucho más fácil para crear versiones localizadas [36]. Los recursos de cadenas se especifican mediante la cadena de etiqueta como se muestra en el siguiente fragmento de XML:

```
<string name="stop_message">Stop.</string>
```

Android soporta el estilo de texto simple, así que se puede usar el HTML tags <b>, <i>, y <u> para aplicar negrita, cursiva o subrayado a las partes de las cadenas de texto [33] como se muestra en el siguiente ejemplo:

```
<string name="stop_message"><b>Stop.</b></string>
```

Se puede utilizar las cadenas de recursos como parámetros de entrada para el método `string.format`. Sin embargo, `string.format` no es compatible con el estilo del texto que se describe anteriormente. Para aplicar un estilo a una cadena de formato, se tiene que escapar de las etiquetas HTML al crear el recurso [36], como se muestra a continuación:

```
<string name="stop_message">&lt;/b>.%1s</string>
```

Dentro del código, se usa el método `Html.fromHtml` para convertir nuevamente en un estilo de secuencia de caracteres:

```
stringrString = getString(R.string.stop_message);  
stringrString = String.format(rString, "Collaborate and listen.");  
CharSequencestyledString = Html.fromHtml(fString);
```

## Colors

Se usa la etiqueta de color para definir un recurso de nuevo color. Se especifica el valor del color con el símbolo `#` seguido por el canal opcional alfa [36]. Los valores de rojo, verde y azul, con uno o dos números hexadecimales con cualquiera de las siguientes notaciones: `#RGB`, `#RRGGBB`, `#ARGB`, `#ARRGGBB`.

## Dimensions

Las dimensiones son más comunes hacer referencia a los recursos de estilo y diseño [36]. Son útiles para la creación de las constantes de diseño, como las fronteras y la altura de la tipografía.

Para especificar un recurso `Dimensions`, se usa la etiqueta `dimensions`, especificando el valor de la dimensión, seguido por una identificación describiendo la magnitud de su dimensión [33]:

- px Screen pixels.
- in Physical inches.
- pt Physical points.
- mm Physical millimeters.
- dp Density-independent pixels relativos a 160-dpi screen.
- sp Scale-independent pixels.

Estas alternativas permiten definir una dimensión no sólo en términos absolutos, sino también con escalas relativas que dan cuenta de diferentes resoluciones de pantalla y densidades para simplificar la ampliación en un hardware diferente.

### **Styles and Themes**

Los recursos de Estilo permiten a las aplicaciones mantener un aspecto coherente al especificar los valores de atributo utilizado por los visitantes. El uso más común de temas y estilos es la de almacenar los colores y fuentes para una aplicación. Se puede cambiar fácilmente la apariencia de la aplicación, solo es necesario especificar un estilo diferente [33].

Para crear un estilo, se usa una etiqueta de estilo que incluye un atributo de nombre, y contiene una o más etiquetas de elemento. Cada etiqueta de elemento debe incluir un atributo de nombre que se utiliza para especificar el atributo [33] (por ejemplo, tamaño de letra o color) que se definió.

### **Utilización de Recursos**

Así como los recursos que crean, Android suministra varios recursos del sistema que se puede utilizar en sus aplicaciones. Los recursos se pueden utilizar directamente desde el código de la aplicación y también se puede hacer referencia a los recursos [34].

Es importante tener en cuenta que al utilizar los recursos no se puede elegir una determinada versión especializada. Android seleccionará automáticamente el valor más

apropiado para un identificador de recursos dada en función del hardware actual y la configuración del dispositivo.

### Usando Recursos en Código

Para acceder a los recursos en el código utilizando la clase `R` estática. `R` es una clase generada sobre la base de los recursos externos y creados mediante la compilación de su proyecto. La clase `R` contiene las subclases estáticas para cada uno de los tipos de recursos para el que se ha definido al menos un recurso [36]. Por ejemplo, el proyecto de nueva disposición predeterminada contiene las subclases `R.string` y `R.drawable`.

Si se utiliza el ADT plug-in de Eclipse, la clase `R`, se creará automáticamente al realizar cualquier cambio, si no se utiliza el ADT plug-in se utilizar la herramienta AAPT que compila el proyecto y genera la clase `R` [33].

Cada una de las subclases dentro de `R` expone sus recursos asociados, como variables, los nombres de variables coincide con el ERS recurso de identificación por ejemplo: `R.string.app_name` ó `R.drawable.icon`. El valor de estas variables es una referencia a la ubicación del recurso correspondiente en la tabla de recursos, no una instancia del recurso en sí mismo. Cuando un constructor o método, como `setContentView`, acepta un recurso de identificación de ERS, se puede pasar a la variable de recursos [33].

Si se necesita una instancia del recurso en sí mismo, se tendrá que utilizar los métodos de ayuda para extraer de la tabla de recursos, representada por una instancia de la clase de recursos.

Debido a que estos métodos realizan búsquedas en la mesa de recursos de la aplicación, estos métodos de ayuda no pueden ser estáticos. Se usa el método `getResources` en su contexto de aplicación, para acceder a instancia de la aplicación de recursos [33]:

```
Resources myResources = getResources();
```

La clase incluye captadores de recursos para cada uno de los tipos de recursos disponibles.

## Lista de Recursos

En la tabla 4.3 se puede observar una lista de recursos y donde situarlos [33], [34].

**Tabla. 4.3. Lista de Recursos y su Dirección.**

Recursos	Dirección
Layout	“/res/layout/”
images	“/res/drawable/”
animations	“/res/anim/”
styles, strings and arrays: <ul style="list-style-type: none"> <li>▪ ‘arrays.xml’ para definir arrays</li> <li>▪ ‘colors.xml’ para definir colors</li> <li>▪ ‘dimens.xml’ para definir dimensions</li> <li>▪ ‘strings.xml’ para definir strings</li> <li>▪ ‘styles.xml’ para definir style objects</li> </ul>	“/res/values/”
raw files como mp3 ó videos	“/res/raw/”

## Referenciando Recursos

Se puede hacer referencia a los recursos para su uso como valores de los atributos de otros recursos de XML. Esto es particularmente útil para los diseños y estilos, lo que permite crear variaciones sobre temas especializados, cadenas localizadas y gráficos. También es una forma útil de apoyo a las diferentes imágenes y el espacio para un diseño, para asegurarse de que está optimizado para diferentes tamaños de pantalla y resoluciones. Para hacer referencia a un recurso de otro, se utiliza la notación @ [34].

### 4.3.6 Ciclo de vida de una Aplicación Android

A diferencia de la mayoría de los entornos tradicionales, las aplicaciones de Android no tienen control sobre su propio ciclo de vida. En lugar de ello, los componentes de la aplicación deben escuchar los cambios en el estado de la aplicación y reaccionar rápidamente, pero en especial estar preparados para la terminación prematura o repentina. Como se mencionó antes, cada aplicación Android se ejecuta en su propio proceso que está corriendo. La memoria y la gestión de procesos de cada aplicación son manejadas exclusivamente por el tiempo de ejecución [33].

Aunque es poco frecuente, es posible forzar a los componentes de aplicación en la misma aplicación que se ejecutan en procesos diferentes o tener múltiples aplicaciones que comparten el mismo proceso.

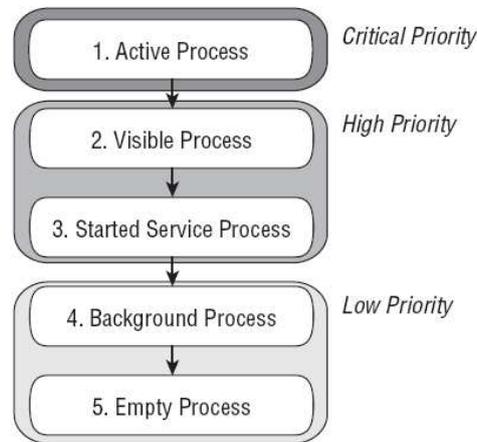
Android administra sus recursos, haciendo todo lo posible para asegurar que el dispositivo sigue siendo sensible. Esto significa que los procesos y sus aplicaciones hospedadas serán terminados, sin previo aviso, y si es necesario libera recursos para las aplicaciones de mayor prioridad, por lo general los que están interactuando directamente con el usuario en el momento [33].

#### **Prioridad de Aplicaciones y Estado de Procesos**

El orden en que los procesos estén cerrados para reclamar los recursos está determinado por la prioridad de las aplicaciones alojadas. Una prioridad de aplicación es igual a su más alta prioridad de los componentes.

Cuando las dos aplicaciones tienen la misma prioridad, el proceso que ha sido una prioridad más baja, morirá primero. La prioridad del proceso también se ve afectada por las dependencias entre procesos, si una aplicación tiene una dependencia en un servicio o proveedor de contenidos suministrados por una segunda aplicación, la aplicación secundaria tendrá al menos una prioridad tan alta como la aplicación que soporta [33], como se observa en la Figura 4.8.

Todas las aplicaciones de Android se mantendrán en ejecución y en la memoria hasta que el sistema necesite sus recursos para otras aplicaciones.



**Figura. 4.8. Árbol de prioridad de Procesos.**

Es importante estructurar la aplicación correcta para asegurar que su prioridad es apropiada para el trabajo que se está haciendo. Si no lo hace, la solicitud podría ser terminada, en medio de algo importante.

La siguiente lista detalla cada uno de los estados de la aplicación que se muestra en la Figura 4.9, explicando el estado de cómo está determinado por los componentes de aplicación que lo componen:

- 1. Active Processes:** Activa en Foreground, los procesos son las aplicaciones de hosting con componentes y actualmente interactúan con el usuario. Los procesos de Android se mantienen hasta que se haga una recuperación de los recursos [33]. En general, existen muy pocos procesos que se los termina sólo como último recurso.

En Active processes incluye:

- Las actividades en un estado "activo", es decir, que están en Foreground y responden los eventos del usuario.
- Actividades, Servicios, o receptores de radiodifusión que se están ejecutando actualmente con un controlador de eventos onReceive.

- Servicios que se está ejecutando en `onStart`, `onCreate`, o en el controlador de eventos `OnDestroy`.
2. **Visible Processes:** Procesos visibles, pero inactivos, como el nombre sugiere, las actividades visibles son visibles, pero no están en primer plano o responden a eventos de usuario. Esto ocurre cuando una actividad parcialmente es oscurecido (actividad no en pantalla completa o transparente) [33]. En general, existen muy pocos procesos visibles, que sólo van a terminar en circunstancias extremas, para permitir que los procesos activos continúen.
  3. **Started Service Processes:** Los procesos de servicios de alojamiento son servicios de apoyo de procesamiento en curso, que debe continuar sin una interfaz visible. Debido a que los servicios no interactúan directamente con el usuario, reciben una prioridad ligeramente inferior a las actividades visibles [33]. Se sigue considerando que los procesos `Foreground` no se terminan a menos que los recursos sean requeridos para los procesos activos o visibles.
  4. **Background Processes:** Los procesos de las actividades de alojamiento que no son visibles y que no cuentan con los servicios que se han iniciado, se consideran los procesos de `Background` o de fondo [33]. Suele haber un gran número de procesos de fondo que Android va a terminar con esta frase “el ultimo visto, el primer terminado” patrón para obtener recursos para los procesos de primer plano `Foreground`.
  5. **Empty Processes:** Para mejorar el rendimiento general del sistema, Android puede ejercer las aplicaciones en la memoria después de haber llegado al final de su vida. Android mantiene esta memoria caché para mejorar el tiempo de puesta en marcha de aplicaciones cuando se reinician [33]. Estos procesos son rutinariamente terminados cuando sea necesario.

### 4.3.7 Herramientas para el desarrollo en Android

El SDK de Android incluye varias herramientas y utilidades para ayudarle a crear, probar y depurar sus proyectos.

Como se mencionó anteriormente, el ADT plug-in convenientemente incorpora la mayoría de estas herramientas en el Eclipse IDE, donde se puede acceder a ellas desde el punto de vista DDMS, incluyendo [33]:

- **El SDK de Android y el Administrador de dispositivos virtuales:** Utilizados para crear y administrar Android Virtual Devices (AVD) y paquetes SDK. EL AVD alberga un emulador para ejecutar una determinada construcción de Android, lo que le permite especificar la versión del SDK de apoyo, resolución de pantalla, la cantidad de almacenamiento de tarjeta SD disponibles, y las capacidades de hardware disponibles (por ejemplo, pantallas táctiles y GPS).
- **Android Emulator:** Es una implementación de la máquina virtual de Android diseñado para ejecutarse en un dispositivo virtual en el equipo de desarrollo [36]. Se usa el emulador para probar y depurar las aplicaciones Android.
- **Depuración Dalvik Servicio de Monitoreo (DDMS):** Se usa DDMS para supervisar y controlar las Máquinas Virtuales Dalvik en el que se está depurando las aplicaciones.
- **Android Packaging Tool Actives (AAPT):** Construye el paquete distribuable de archivos de Android (.apk).
- **Android depuración Bridge (ADB):** Es una aplicación cliente-servidor que proporciona un enlace a un emulador en funcionamiento, permitiendo copiar archivos, instalar paquetes de aplicaciones compiladas (.apk), y ejecutar comandos del Shell.

Herramientas adicionales disponibles [33]:

- **SQLite3:** Es una herramienta de base de datos que se pueden utilizar para acceder a los archivos de base de datos SQLite creado y usado por Android, se explicará en la sección 4.4.7.

- **Traceview:** Es una herramienta de análisis gráfico para visualizar los registros de seguimiento de su aplicación Android.
- **MkSDCard:** Crea una tarjeta SD de imagen de disco que puede ser utilizado por el emulador para simular una tarjeta de almacenamiento externa.
- **Dx:** Convierte clases de Java Bytecode en AndroidBytecode.dex.
- **ActivityCreator:** Se utiliza para compilar las aplicaciones Android sin el ADT plug-in.
- **LayoutOpt:** Es una herramienta que analiza los recursos de diseño y propone mejoras y optimizaciones.

## Android Virtual Devices

Android Virtual Devices se utiliza para simular las compilaciones de software y las especificaciones de hardware disponibles en diferentes dispositivos. Esto permite probar la aplicación en una variedad de plataformas de hardware sin necesidad de comprar una variedad de teléfonos [33]. Cada dispositivo virtual está configurado con un nombre, un destino de generación de Android (basado en la versión SDK compatible), con una capacidad de tarjeta SD, y la resolución de la pantalla [33], como se muestra en Figura 4.9.



Figura. 4.9. Ventana de Nuevo AVD.

Cada dispositivo virtual también es compatible con un número de configuraciones de hardware y restricciones específicas que se pueden agregar en forma de NVPs en la tabla de hardware. Estas opciones adicionales incluyen [33]:

- Máximo tamaño de la pila de la máquina virtual.
- Densidad de píxeles de la pantalla.
- SD tarjeta de apoyo.
- La existencia de pantalla táctil, teclado y trackball hardware.
- Acelerómetro y compatibilidad con GPS.
- Memoria disponible del dispositivo.
- Cámara de hardware y resolución.
- Soporte para la grabación de audio.

Diferentes configuraciones de hardware y resoluciones de pantalla presentan alternativas de interfaz de aspectos de usuario para representar las diferentes configuraciones de hardware. Esto simula una gran variedad de tipos de dispositivos móviles. Para completar la ilusión, se puede crear un aspecto personalizado para cada dispositivo virtual, para hacer que parezca que el dispositivo se está emulando [33].

### **SDK Manager**

Se usa las fichas de paquetes instalados y disponibles para administrar las instalaciones de SDK. Los paquetes instalados, que se muestra en la Figura 4.10, muestran las plataformas SDK, documentación y herramientas que se tiene disponible para su uso en el entorno de desarrollo.

Para actualizar a una nueva versión simplemente se hace clic en el botón Update All para que el gestor de actualizaciones de la instalación de SDK actualice a la última versión de cada componente [33], como se observa en la Figura 4.10.

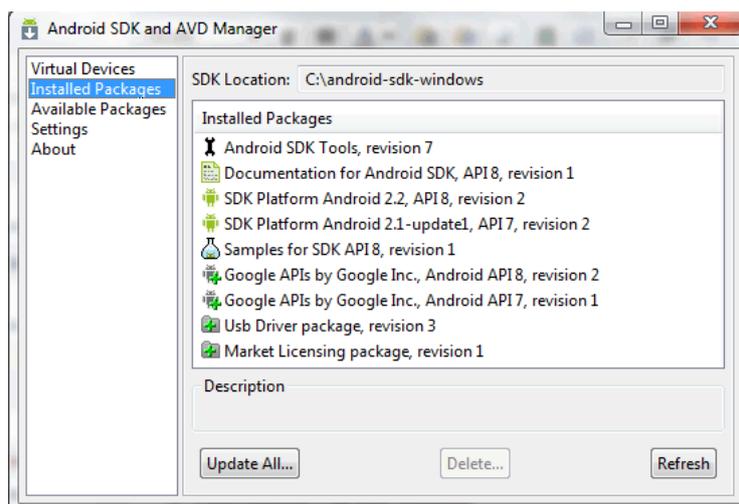


Figura. 4.10. Android SDK y AVD Manager –Instalación de Paquetes.

Por otra parte, los paquetes disponibles, comprueban el repositorio de Android SDK para cualquier fuente, paquetes, y los archivos disponibles, que aún no están instalados en el sistema. Se usa las casillas de verificación, como se muestra en la Figura 4.11, para seleccionar paquetes adicionales para instalar SDK [33].

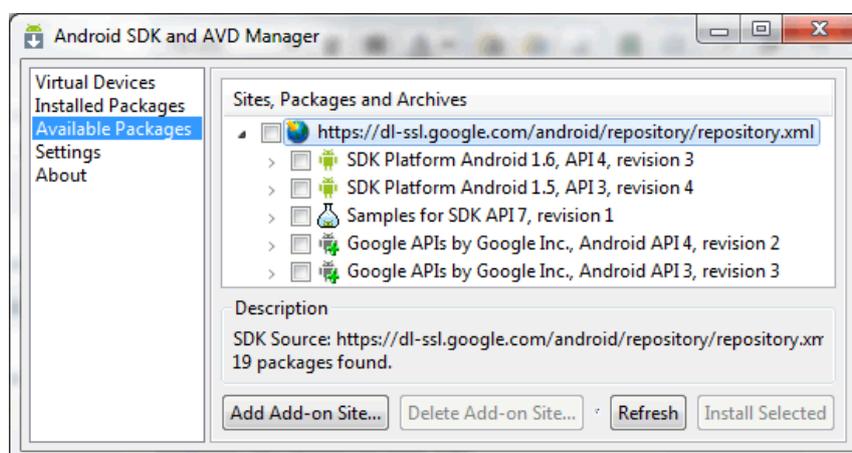


Figura. 4.11. Casillas de Paquetes disponibles.

## Android Emulator

El Emulador es la herramienta perfecta para probar y depurar las aplicaciones. Es una implementación de la máquina virtual Dalvik, por lo que es válido como una plataforma para correr aplicaciones Android, como cualquier otro teléfono Android. Debido a que es disociado de cualquier hardware en particular, es una base excelente para usar y probar las aplicaciones. Proporciona plena conectividad de red, junto con la

habilidad de ajustar la velocidad de conexión a Internet y la latencia durante la depuración de sus aplicaciones. También se puede simular realizando y recibiendo llamadas de voz y mensajes SMS [33].

El ADTplug-in se integra al emulador de Eclipse para que sea enganchado de forma automática dentro de la AVD seleccionando ejecutar o depurar sus proyectos. Para ejecutar el emulador primero se tiene que crear un dispositivo virtual. El emulador iniciará el dispositivo virtual y ejecutará una instancia de Dalvik dentro de ella [33].

### **Dalvik Debug Monitor Service (DDMS)**

El emulador permite ver cómo se verá la aplicación, su comportamiento e interacción, pero para ver realmente lo que está sucediendo bajo la superficie se necesita la depuración Dalvik Debug Monitoring Service. El DDMS es una herramienta de depuración de gran alcance que permite interrogar a los procesos activos, ver la pila y el montón de temas, ver y hacer una pausa activa, y explorar el sistema de archivos de cualquier dispositivo conectado a Android.

La perspectiva DDMS en Eclipse también ofrece un acceso simplificado a las capturas de pantalla del emulador y los registros generados por LogCat.

Si se está utilizando el ADT plug-in de, la DDMS está totalmente integrada en Eclipse y está disponible desde la perspectiva DDMS. Si no se está utilizando el plug-in de Eclipse, se puede ejecutar DDMS desde la línea de comandos y se conectará automáticamente a cualquier dispositivo que se ejecute o se emule [33].

### **LogCat**

El LogCat es una parte de la DDMS que proporciona un mecanismo para recoger y visualizar la salida de depuración del sistema en tiempo real y en conjunto con el dispositivo móvil inteligente. Los registros de diversas aplicaciones y partes del sistema se recogen en el LogCat, que se pueden ver y filtrar [34].



Como un conducto de comunicación entre el hardware y el desarrollo del dispositivo Android / emulador, el ADB permite instalar aplicaciones, ejecutar y parar archivos, y ejecutar comandos de shell en el dispositivo de destino. Utilizando el dispositivo de shell se puede cambiar la configuración de registro, y consultar o modificar bases de datos SQLite en el dispositivo [33].

La herramienta ADT automatiza y simplifica mucho la interacción habitual con el ADB, incluida la instalación de la aplicación y actualización, el archivo de registro y transferencia de archivos [33] (a través de la perspectiva DDMS).

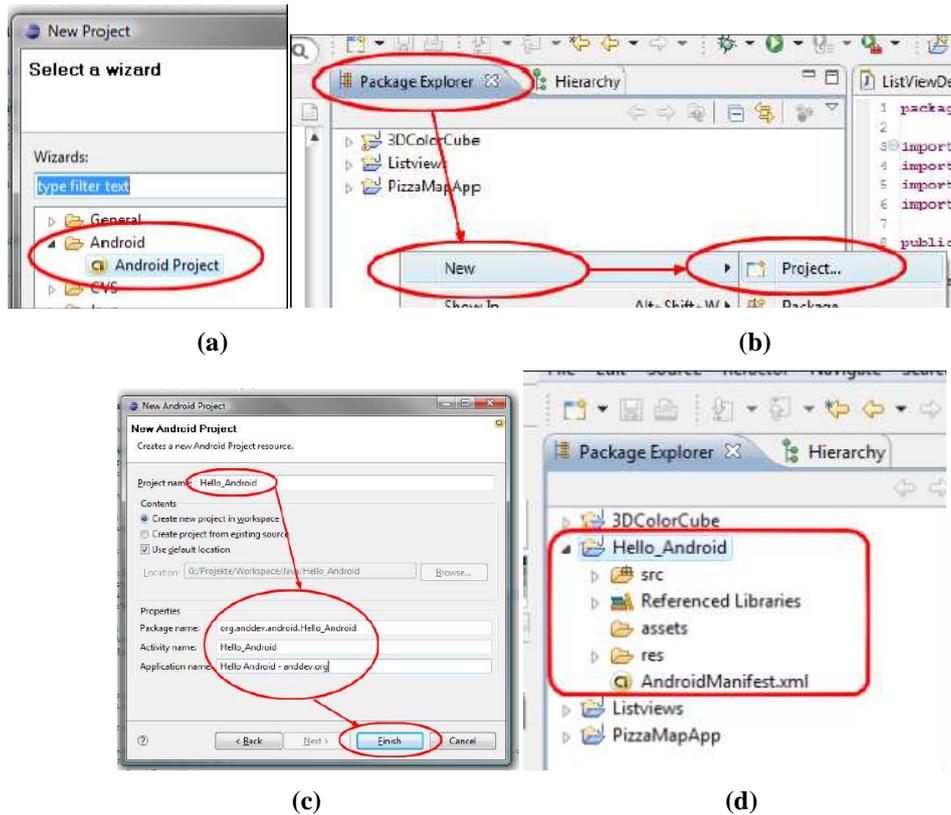
## **4.4 DESARROLLO DE APLICACIONES ANDROID**

Si ya se ha descargado el SDK, instalado Eclipse, y se conecta en el ADT plug-in. Se está listo para comenzar a programar para Android. Se comienza por crear un nuevo proyecto, la configuración de su ejecución y las configuraciones de depuración de Eclipse.

### **4.4.1 Creación de un nuevo proyecto de Android**

Para crear un nuevo proyecto Android se utiliza el asistente para nuevo proyecto de Android [34] siguiendo los siguientes pasos:

1. Se selecciona Archivo ⇨ Nuevo ⇨ Proyecto.
2. Se toma el proyecto Android, tipo de aplicación de la carpeta de Android, y se hace clic en finalizar.
3. En el cuadro de diálogo que aparece (como se muestra en la Figura 4.13), se ven los detalles del nuevo proyecto. El "Nombre del proyecto", el "Nombre del paquete" es específicamente el paquete, el "Nombre de la actividad" es el nombre de la clase que es la actividad inicial, y el "Nombre de aplicación" es el nombre para la aplicación a desarrollar.
4. Cuando se haya introducido los datos, se hace clic en Finalizar.



**Figura. 4.13. Primera Aplicación Android.**

(a) Paso 1. (b) Paso 2. (c) Paso 3. (d) Paso 4.

El ADT plug-in crea un nuevo proyecto que incluye una nueva clase que se extiende a la actividad. En lugar de estar completamente vacío, la plantilla predeterminada implementa "Hello World" antes de modificar el proyecto [34].

### Creación de una Configuración Inicial.

Las configuraciones de Inicio permiten especificar las opciones de tiempo de ejecución para ejecutar y depurar aplicaciones. El uso de una configuración inicial puede especificar lo siguiente:

- El proyecto y la actividad de lanzamiento.
- Las opciones de emulador a usar.
- Entrada / salida de la configuración (incluido por defecto de la consola).

Se puede especificar diferentes configuraciones iniciales de confianza para ejecutar y los modos de depuración. Los pasos siguientes muestran cómo crear una configuración inicial para una aplicación de Android [34]:

1. Seleccionar Ejecutar ⇨ diálogo Open Run (o de diálogo Ejecutar ⇨ depuración abierta).
2. Se hace clic derecho en la aplicación Android en la lista el tipo de proyecto, y se selecciona nuevo.
3. Se escribe un nombre para la configuración. Luego se crea varias configuraciones de confianza para cada proyecto, por lo que se debe crear un título descriptivo que ayudará a identificar esta configuración en particular.
4. Elegir la puesta en marcha de opciones. La primera pestaña permite seleccionar el proyecto y actividad que se desea iniciar cuando se ejecuta (o depura) de la aplicación.
5. Utilizar la ficha de destino para configurar el emulador. Hay opciones para elegir el tamaño de la pantalla del dispositivo, configuración del skin, y la conexión de red. Opcionalmente, también se puede borrar los datos del usuario en el emulador y activar o desactivar la animación de inicio.
6. Utilizando el cuadro de texto de línea de comandos, se puede especificar la configuración del emulador con más opciones de inicio si es necesario.
7. Por último, se establece las propiedades adicionales en la ficha común.
8. Se hacer clic en aplicar, y la configuración de inicialización se guardará.

### **Ejecutar y depurar aplicaciones Android**

Hasta ahora se ha creado un primer proyecto con la configuración de depuración y configuración de iniciación de confianza. Para verificar si todo esta correcto, antes de hacer cualquier cambio, se debe probar la instalación y configuraciones por ejecutar y depurar el proyecto *“Hello World”* [34].

En el menú Ejecutar, se selecciona Ejecutar o Depurar, para iniciar o seleccionar abrir el diálogo Ejecutar. El ADT plug-in, en ejecución o depuración de la aplicación realiza los siguientes procesos:

- Compila el proyecto actual y lo convierte en un archivo ejecutable Android (.dex).
- Crea los Paquetes de los recursos ejecutables y externos en un paquete de Android (.apk).
- Inicia el emulador.
- Se instala la aplicación en el emulador.
- Inicia su aplicación.

Si todo funciona correctamente, se verá una nueva ventana de actividad en el emulador, como se muestra en la Figura 4.14.



**Figura. 4.14. Ventana de Emulador.**

### **Corriendo una aplicación Android**

Se abre el menú desplegable en el menú superior de Eclipse y se hace clic en Abrir cuadro de diálogo y Ejecutar como se ve en la Figura 4.15.

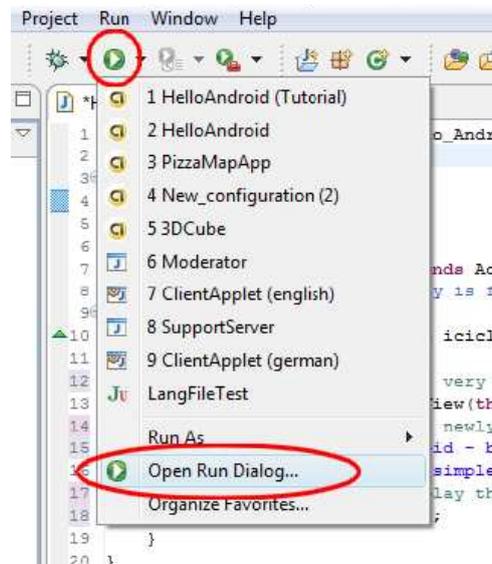


Figura. 4.15. Run-Dialog.

Aparece una ventana de Create, manage, y runconfigurations en donde se puede configurar el Nombre del Proyecto y escoger la actividad, como se muestra en la Figura 4.16.

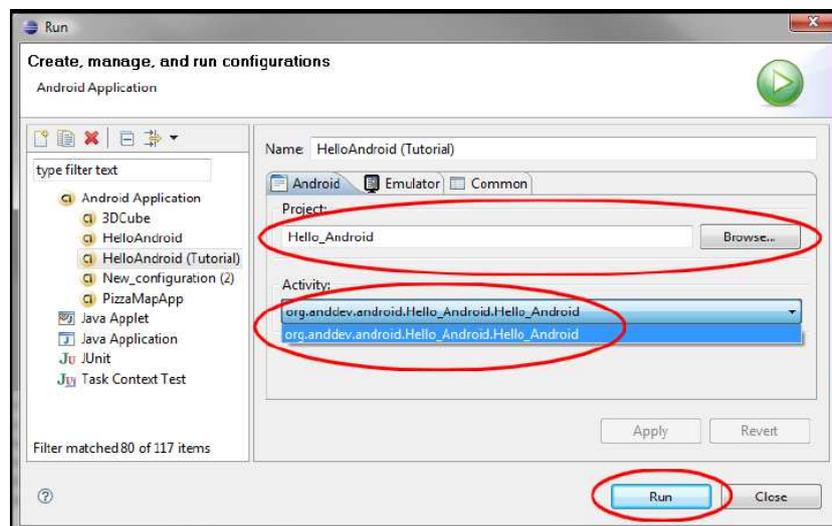


Figura. 4.16. Run Configuration.

A partir de ahora se puede ejecutar la aplicación haciendo clic en el botón Run. Una vez hecho esto se verá la pantalla del emulador como se muestra en la Figura 4.17.



Figura. 4.17. Primera Android Application.

Después de desplegar la primera aplicación, también se puede ver la aplicación que se encuentra en el menú rápido, con el icono por defecto.

#### 4.4.2 Estructura de un proyecto Android

Cuando se crea un nuevo proyecto Android en Eclipse se genera automáticamente la estructura de carpetas necesaria para poder generar posteriormente la aplicación. Esta estructura será común a cualquier aplicación, independientemente de su tamaño y complejidad.

En la Figura 4.18 se ve los elementos creados inicialmente para un nuevo proyecto Android:



Figura. 4.18. Elementos creados inicialmente para un nuevo proyecto Android.

## Elementos principales

### Carpeta /src/

En la Figura 4.19 se muestra el contenido de la carpeta /src/ la cual contiene todo el código fuente de la aplicación, código de la interfaz gráfica, clases auxiliares, etc. Inicialmente, Eclipse creará el código básico de la pantalla (Activity) principal de la aplicación, siempre bajo la estructura del paquete java definido [33], [34].



Figura. 4.19. Subcarpetas de /src/.

### Carpeta /res/

Contiene todos los ficheros de recursos necesarios para el proyecto: imágenes, vídeos, cadenas de texto, etc [33], [34]. Los diferentes tipos de recursos se deberán distribuir entre las siguientes carpetas que muestra la Tabla 4.4.

Tabla. 4.4. Subcarpetas de /res/.

Propiedades	Función
/res/drawable/	Contienen las imágenes de la aplicación. Se puede dividir en /drawable-ldpi, /drawable-mdpi y /drawable-hdpi para utilizar diferentes recursos dependiendo de la resolución del dispositivo.
/res/layout/	Contienen los ficheros de definición de las diferentes pantallas de la interfaz gráfica. Se puede dividir en /layout y /layout-land para definir distintos layouts dependiendo de la orientación del dispositivo.
/res/anim/	Contiene la definición de las animaciones utilizadas por la aplicación.
/res/menu/	Contiene la definición de los menús de la aplicación.
/res/values/	Contiene otros recursos de la aplicación como por ejemplo cadenas de texto (strings.xml), estilos (styles.xml), colores (colors.xml), etc.
/res/xml/.	Contiene los ficheros XML utilizados por la aplicación.

En la Figura 4.20 se muestran los siguientes recursos creados para la aplicación en un nuevo proyecto Android.

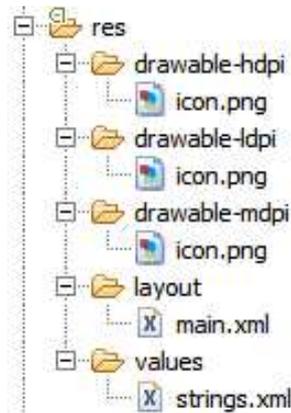


Figura. 4.20. Recursos de la Aplicación.

### Carpeta /gen/

Contiene una serie de elementos de código generados automáticamente al compilar el proyecto. Cada vez que se genera el proyecto, la maquinaria de compilación de Android genera una serie de ficheros fuente en java, dirigidos al control de los recursos de la aplicación [33], [34] como se muestra en la Figura 4.21.

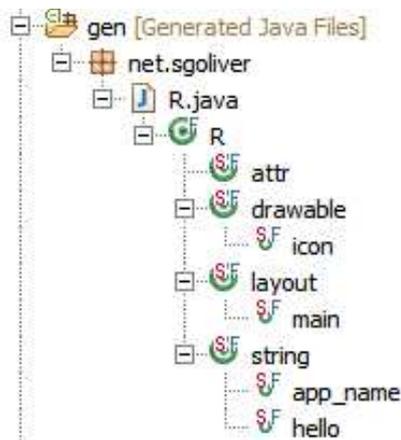


Figura. 4.21. Elementos de código generado automáticamente.

### Clase R.java

El más importante código generado automáticamente es el que se puede observar en el fichero R.java, y la clase R [33], [34].

```
package net.sgoliver;
public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```

Esta clase R contiene en todo momento una serie de constantes con los ID de todos los recursos de la aplicación incluidos en la carpeta /res/, de forma que podamos acceder fácilmente a estos recursos desde nuestro código a través de este dato. Así, por ejemplo, la constante R.drawable.icon contendrá el ID de la imagen “icon.png” contenida en la carpeta /res/drawable/.

### **Carpeta /assets/**

Contiene todos los demás ficheros auxiliares necesarios para la aplicación (y que se incluirán en su propio paquete), como por ejemplo ficheros de configuración, de datos, etc.

La diferencia entre los recursos incluidos en la carpeta /res/raw/ y los incluidos en la carpeta /assets/es, que para los primeros se generará un ID en la clase R y se deberá acceder a ellos con los diferentes métodos de acceso a recursos. Para los segundos sin embargo no se generarán ID y se podrá acceder a ellos por su ruta como a cualquier otro fichero del sistema. Se usa uno u otro según las necesidades de la aplicación [33], [34].

### **4.4.3 Creación de un Activity**

Dentro de esta nueva clase se debe definir la interfaz de usuario e implementar su funcionalidad. El código esqueleto básico de una nueva actividad se muestra a continuación [33].

```
Package com.paad.myapplication;

importAndroid.app.myapplication;
importandroid.os.Bundle;

public class MyActivity extends Activity {

    /**called when the activity is first created. */
    @Override
    Public void onCreate(Bundle savedInstanceState) {
        Super.onCreate(savedInstanceState);
    }
}
```

La base de la clase Activity presenta una pantalla en blanco que encierra el manejo de visualización de la ventana. Un activismo vacío y no es particularmente útil, por lo que la primera cosa que se debe hacer es crear la interfaz de usuario con vistas y diseños.

Las vistas son los controles de la interfaz de usuario que muestran los datos y proporcionar la interacción del usuario. Android proporciona varias clases de diseño, llamada View Groups, que puede contener múltiples puntos de vista para ayudar a diseñar las interfaces de usuario. Para asignar una interfaz de usuario de un Activity, se llama al método setContentView de onCreate de su Activity. Para utilizar una actividad en la aplicación. Se realiza los siguientes pasos [33]:

Agregar nuevas etiquetas <activity> dentro del nodo <application> del manifest, la etiqueta <activity> incluye los atributos de los metadatos, como la etiqueta, icono, los permisos necesarios, y los temas utilizados por la actividad. Una actividad sin la correspondiente etiqueta <activity> no se puede mostrar.

#### 4.4.4 Creación de una Interfaz de Usuario

Es vital que se cree interfaces de usuario atractivas e intuitivas para las aplicaciones. Además asegurar que sean lo más elegante y fácil de usar, ya que son funcionales que deben ser una prioridad de diseño superior.

Los elementos de una interfaz de usuario de Android están dispuestos en la pantalla por medio de una variedad de controladores de distribución derivados de View Group. El uso correcto de los diseños es esencial para la creación de interfaces [33].

### **Diseño de una UI Android**

Android introduce una nueva terminología para las metáforas de programación que se analizarán en detalle en las siguientes secciones:

- **Views:** Son la clase base para todos los elementos de la interfaz visual (comúnmente conocidos como controles o Widgets) [33]. Todos los controles de la interfaz de usuario, incluidas las clases de diseño, se derivan del view.
- **View Groups:** son extensiones de la clase View que puede contener múltiples puntos de vista. Ampliación de la clase ViewGroup para crear controles compuestos, formados por view[33]. La clase ViewGroup se extiende también a ofrecer a los gestores de diseño que ayudan a diseñar los controles dentro de las actividades.
- **Activities:** descrito en detalle en el capítulo anterior, representan la ventana o pantalla que se muestra. Las actividades son el equivalente Android de las Forms [33]. Para mostrar una interfaz de usuario se asigna una vista (por lo general un diseño) para una actividad.

Android proporciona varios controles de interfaz de usuario común, Widgets, y los administradores de disposición. Para la mayoría de las aplicaciones gráficas es muy probable que se tenga que ampliar y modificar el presente dictamen estándar o crear vistas compuestas o totalmente nuevas para ofrecer su experiencia de usuario.

### **Capas o Layouts**

Los Layouts son elementos no visuales destinados a controlar la distribución, posición y dimensión de los controles que se insertan en su interior. Estos componentes extienden a la clase base ViewGroup, como muchos otros componentes contenedores, es decir, capaces de contener a otros controles.

## Frame Layout

Éste es el más simple de todos los Layouts de Android. Un Frame Layout coloca todos sus controles hijos alineados con su esquina superior izquierda, de forma que cada control quedará oculto por el control siguiente (a menos que éste último tenga transparencia). Por ello, suele utilizarse para mostrar un único control en su interior, a modo de contenedor (placeholder) sencillo para un sólo elemento sustituible, por ejemplo una imagen [33].

Los componentes incluidos en un Frame Layout podrán establecer sus propiedades `android:layout_width` y `android:layout_height`, que podrán tomar los valores “`fill_parent`” (para que el control hijo tome la dimensión de su layout contenedor) o “`wrap_content`” [33] (para que el control hijo tome la dimensión de su contenido). En la Figura 4.22 se puede ver un ejemplo de un Frame Layout.



Figura. 4.22. Frame Layout.

## Linear Layout

El siguiente Layout Android en cuanto a nivel de complejidad es el Linear Layout. Este Layout apila uno tras otro todos sus elementos hijos de forma horizontal o vertical según se establezca su propiedad `android:orientation`.

Al igual que en un Frame Layout, los elementos contenidos en un Linear Layout pueden establecer sus propiedades `android:layout_width` y `android:layout_height` para determinar sus dimensiones dentro del Layout. Pero en el caso de un Linear Layout,

tendremos otro parámetro con el cual podemos configurar, la propiedad `android:layout_weight` [33].

A pesar de la simplicidad aparente de este Layout resulta ser lo suficiente versátil como para ser de utilidad en muchas ocasiones [33]. Se puede ver un ejemplo de un Linear Layout en la Figura 4.23.

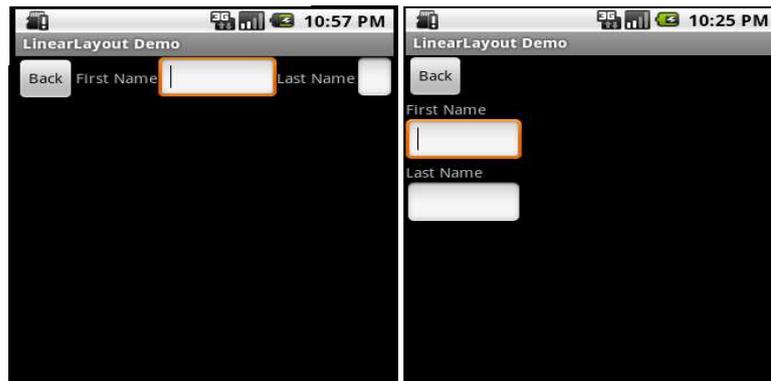


Figura. 4.23. Linear Layout.

## Table Layout

Un Table Layout permite distribuir sus elementos hijos de forma tabular, definiendo las filas y columnas necesarias, además la posición de cada componente dentro de la tabla.

La estructura de la tabla se define de forma similar a como se hace en HTML, es decir, indicando las filas que compondrán la tabla (objetos `TableRow`), y dentro de cada fila las columnas necesarias, con la salvedad de que no existe ningún objeto especial para definir una columna (similar a un `TableColumn`) sino que directamente se inserta los controles necesarios dentro del `TableRow` y cada componente insertado (que puede ser un control sencillo o incluso otro `ViewGroup`) corresponderá a una columna de la tabla. De esta forma, el número final de filas de la tabla corresponderá con el número de elementos `TableRow` insertados, y el número total de columnas quedará determinado por el número de componentes de la fila que más componentes contenga [33].

Por norma general, el ancho de cada columna corresponderá con el ancho del mayor componente de dicha columna, pero existe una serie de propiedades que ayudará a modificar este comportamiento. En la Tabla 4.5 se muestra las principales propiedades del Table Layout.

**Tabla. 4.5. Propiedades de Table Layout.**

Propiedades	Función
android:stretchColumns	Indica las columnas que se pueden expandir para absorber el espacio libre dejado por las demás columnas a la derecha de la pantalla.
android:shrinkColumns	Indica las columnas que se pueden contraer para dejar espacio al resto de columnas que se puedan salir por la derecha de la pantalla.
android:collapseColumns	Indica las columnas de la tabla que se quieren ocultar completamente.

Todas estas propiedades del Table Layout pueden recibir una lista de índices de columnas separados por comas (ejemplo: *android:stretchColumns="1,2,3"*) o un asterisco para indicar que se debe aplicar a todas las columnas (ejemplo: *android:stretchColumns="\*"*)[33].

## Relative Layout

Este Layout permite especificar la posición de cada elemento de forma relativa a su elemento padre o a cualquier otro elemento incluido en el propio Layout [33]. De esta forma, al incluir un nuevo elemento X se podrá indicar por ejemplo que se debe colocar debajo del elemento Y alineado a la derecha del Layout padre.

En un Relative Layout se tendrá un sinnúmero de propiedades para colocar cada control justo donde se requiere como se muestra en la Figura 4.24.



**Figura. 4.24. Relative Layout.**

#### 4.4.5 Utilización de Intents

Los Intents son utilizados como un mecanismo de paso de mensajes que funciona tanto en su aplicación, y entre otras aplicaciones. Los Intents se pueden utilizar para:

- Declarar su intención de que una actividad o servicio comenzó a realizar una acción, por lo general con una parte de los datos.
- Difusión que un evento se ha producido.
- Explícitamente iniciar un servicio o actividad.

Se usa Intents para apoyar la interacción entre cualquiera de los componentes de la aplicación instalada en un dispositivo Android, independientemente de la aplicación. Esto convierte el dispositivo desde una plataforma que contiene una colección de componentes independientes en un único sistema interconectado.

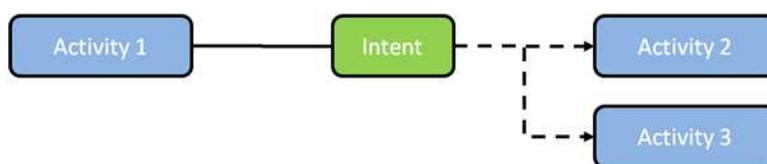
Uno de los usos más comunes de Intents es iniciar nuevas actividades, ya sea explícitamente (mediante la especificación de la clase de carga) o implícitamente [33]. En este último caso, la acción no tiene que ser realizada por una actividad dentro de la aplicación que llama.

Los Intents también se pueden utilizar para transmitir mensajes a través del sistema. Esto permite crear por eventos aplicaciones basadas en el interno, el sistema, o de otros eventos de aplicación. Las emisiones de Intents de Android permiten anunciar los eventos del sistema, al igual que cambios en el estado de conexión a Internet o los niveles de carga de la batería [33].

#### **Utilización de Intents para llamar a SubActivities**

## SubActivities

Las SubActivities es algo esencial en la vida de una aplicación, que es más sofisticado que una aplicación “*Hello World*”, es empezar otras actividades, especialmente SubActivitites [33], como se muestra en la Figura 4.25.



**Figura. 4.25. Intents y SubActivities.**

## Acciones nativas de Android

Las aplicaciones nativas de Android también utilizan Intents para poner en marcha actividades y sub actividades.

La siguiente Tabla 4.6, muestra algunas de las acciones nativas disponibles como constantes de cadena estática en la clase de Intents. Al crear Intents implícitos se puede utilizar estas acciones, llamado Intenciones de actividad, para iniciar las actividades y subactividades dentro de sus aplicaciones [33], [34].

**Tabla. 4.6. Acciones de Intents.**

Acciones de Intents	Función
ACTION_ANSWER	Abre una actividad que se encarga de las llamadas entrantes. En la actualidad esto es manejado por los nativos en la notificación de llamada en la pantalla.
ACTION_CALL	Muestra un marcador de teléfono y de inmediato inicia una llamada usando el número suministrado en el Intents URI. En general se considera una mejor forma de usar ACTION_DIAL.
ACTION_DELETE	Comienza una actividad que permite borrar los datos especificados en el Intents URI de datos.
ACTION_DIAL	Muestra una aplicación de marcador con el número a marcar antes de la población de la Intents URI. Por defecto esto es manejado por el marcador de teléfono nativo de Android. El marcador puede normalizar la mayoría de los esquemas de número.

ACTION_EDIT	Solicitudes de una actividad que puede editar los datos en el Intents URI especificado.
ACTION_INSERT	Abre una actividad capaz de insertar nuevos elementos en el cursor especificado en el Intents URI. Cuando se llama como una actividad secundaria que debe devolver un URI al elemento recién insertado.
ACTION_SEARCH	Inicia la actividad utilizada para realizar una búsqueda. Suministra los términos de búsqueda como una cadena en la intención de extras con SearchManager.
ACTION_SENDTO	Lanza una actividad para enviar un mensaje al contacto especificado por el Intents URI.
ACTION_SEND	Lanza una actividad que envía los datos especificados en el Intents. El contacto del destinatario debe ser seleccionado por la actividad resuelta. Se usa settype para establecer el tipo MIME de los datos transmitidos.
ACTION_VIEW	La acción genérica más común. View pide que los datos suministrados en el Intents URI puedan ser vistos de manera más razonable. Las diferentes aplicaciones se encargarán de las solicitudes en función de ver el esquema de URI de los datos suministrados.
ACTION_WEB_SEARCH	Abre una actividad que realiza una búsqueda en Internet basado en el texto suministrado en el Intents URI (por lo general el navegador).

### Intent Filters

Los Filtros de Intents se utilizan para registrar las actividades, servicios y receptores de radiodifusión por ser capaz de realizar una acción en un determinado tipo de datos [33], [34].

También se utiliza para registrar los receptores de radiodifusión que están interesados en Intents de difusión, una acción o evento. Utilizar filtros de Intents, anuncia que puedan responder a las solicitudes de acción de cualquier aplicación instalada en el dispositivo.

Para registrar un componente de aplicación como un controlador de Intents potencial, se agrega una etiqueta de filtro de Intents para el nodo Manifest con las siguientes etiquetas (y atributos asociados) en el nodo de filtros de Intents.

- **Action:** Utiliza `android:name` para especificar el nombre de la acción que está en servicio. Cada filtro de Intents debe tener una y sólo una etiqueta de acción [33].

Las acciones deben ser cadenas únicas que se describen a sí mismas. Lo más recomendable es utilizar un sistema de nomenclatura basado en las nomenclaturas de paquetes java.

- **Category:** Se utiliza `android:name` para especificar las circunstancias en que la acción debe ser reparado. Cada etiqueta de filtro Intents puede incluir varias etiquetas de categoría. Puede especificar sus propias categorías o utilizar los valores estándar que proporciona Android [33], enumerados a continuación en la Tabla 4.7.

**Tabla. 4.7. Etiquetas de Category.**

Etiquetas de Category	Función
<b>ALTERNATIVE</b>	En esta categoría se especifica que esta acción debe estar disponible como una alternativa a la acción por defecto a cabo en un elemento de este tipo de datos.
<b>BROWSABLE</b>	Similares a la categoría de alternative, se utiliza cuando una lista de posibilidades es necesario.
<b>DEFAULT</b>	Especifica una acción que está disponible desde el navegador, esta se dispara desde el navegador que siempre incluirá la categoría browsable, por ejemplo, interceptar enlaces a un sitio web en particular.
<b>GADGET</b>	Al establecer la categoría de gadget que especifica que esta actividad puede funcionar embebida dentro de otra actividad.
<b>HOME</b>	Al establecer una categoría de filtro de intents como en casa, sin especificar una acción, se podrá presentar como una alternativa a la pantalla de inicio de origen.

- **Data:** Permite especificar qué tipos de datos de sus componentes puede actuar, se pueden incluir varias etiquetas de datos, según corresponda y utilizar

cualquier combinación de los siguientes atributos vistos en la Tabla 4.8 para especificar los datos del componente [33]:

**Tabla. 4.8. Atributos de Datos.**

Atributos de Datos	Función
<b>android:host</b>	Especifica un nombre de host válido (por ejemplo, google.com).
<b>android:mimeType</b>	Permite especificar el tipo de datos de sus componentes, es capaz de manipularlos.
<b>android:path</b>	Especifica la ruta válida "valores" de la URI (/ et.gr.a, nsport / barcos /).
<b>android:port</b>	Especifica puertos válidos para el host especificado.
<b>android:scheme</b>	Requiere un régimen especial (por ejemplo, el contenido o http).

#### 4.4.6 Trabajando en segundo plano

Android ofrece la clase de servicio para crear componentes de aplicación específicamente para manejar las operaciones y la funcionalidad que se debe ejecutar de forma invisible, sin una interfaz de usuario.

Mediante el uso de los servicios, se puede asegurar que sus aplicaciones continúan ejecutándose y responder a eventos, incluso cuando no estén en uso. Los servicios funcionan sin una interfaz gráfica dedicada, pero, al igual que las actividades y receptores de radiodifusión, se ejecutarán en el hilo principal de la aplicación, el proceso es convirtiéndose en protagonista [34]. Android ofrece varias técnicas para las aplicaciones de comunicación con los usuarios sin una actividad.

#### Servicios

Un servicio es el código de larga duración y se ejecuta sin una interfaz de usuario. Un buen ejemplo de esto es un reproductor multimedia que reproduce las canciones de una lista de reproducción. En una aplicación de reproductor de medios, probablemente no habría una o más actividades que permiten al usuario elegir las canciones y empezar a escucharlas.

Se debe tener en cuenta que se puede conectar a un servicio y ponerlo en marcha si no está ya en ejecución, con el método `Context.bindService()`. Cuando se conecta a un servicio, puede comunicarse con él a través de una interfaz expuesta por el servicio. Para el servicio de música, este podría permitir hacer una pausa, rebobinado, etc.

## Hilos

Un hilo es una unidad de ejecución concurrente. Tiene su propia pila de llamadas a los métodos que se invoca, sus argumentos y variables locales. Cada instancia de máquina virtual tiene al menos un hilo conductor cuando se inicia, normalmente, hay varios de orden interno. Hilos es un tema muy extenso por lo que solo se nombrará lo más relevante acerca de este tema. Hay básicamente dos maneras principales de tener un hilo. Una de ellas es proporcionar una nueva clase del método `run()`. La otra es proporcionar una instancia de mensaje nuevo con un objeto `Runnable` durante su creación [33]. En ambos casos, el método `start()` debe ser llamada para ejecutar realmente el nuevo hilo.

Cada hilo tiene una prioridad entera que básicamente es determinar la cantidad de tiempo por el procesador. Se puede configurar utilizando el `setPriority(int)`. Si se inicia una actividad Android (línea A de la Figura 4.26) y en determinado punto de esta línea, se ejecuta otro hilo (línea B), el resultado serán dos acciones simultáneas en una misma actividad.

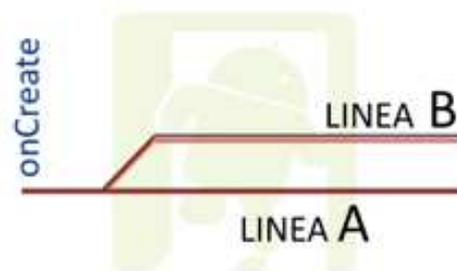


Figura. 4.26. Hilos.

#### 4.4.7 Utilización de bases de datos SQLite y Content Providers

La persistencia de datos estructurados en Android se proporciona a través de los siguientes mecanismos:

- **SQLite Databases:** Cuando se maneja, los datos estructurados es el mejor enfoque, Android ofrece la biblioteca SQLite que es una base de datos relacional. Cada aplicación puede crear sus propias bases de datos sobre los cuales tiene el control completo [33].
- **Content Providers:** Ofrece proveedores de contenido genérico, y una interfaz para utilizar y compartir datos.

SQLite, abarcará todas las tareas relacionadas con el almacenamiento de los datos propios de la aplicación. El segundo de los mecanismos, los Content Providers, facilitarán la tarea de hacer visibles esos datos a otras aplicaciones, y de forma recíproca permitir la consulta de datos publicados por terceros desde la aplicación.

##### SQLite Databases

La plataforma Android proporciona herramientas principales para el almacenamiento y consulta de datos estructurados [33]. SQLite es un motor de bases de datos muy popular en la actualidad por ofrecer características tan interesantes como su pequeño tamaño, no necesitar servidor, precisar poca configuración, ser transaccional y por supuesto 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 es a través de una clase auxiliar llamada SQLiteOpenHelper, o para ser más exactos, de una clase propia que derive de ella y que se debe personalizar para adaptarse a las necesidades concretas de la aplicación [33].

La clase `SQLiteOpenHelper` tiene tan sólo un constructor, que normalmente no se necesita sobrescribir, y dos métodos abstractos, `onCreate()` y `onUpgrade()`, que se debe personalizar con el código necesario para crear la base de datos y para actualizar su estructura respectivamente.

## **Content Providers**

Los proveedores de contenido proporcionan una interfaz para la publicación y los datos de consumo, basado en un modelo sencillo URI, que permiten separar la capa de aplicación de la capa de datos, por lo que sus aplicaciones de datos de código son independientes del ocultamiento del origen de datos subyacente.

Los proveedores de contenidos pueden ser compartidos para ser consultado por los resultados, los registros existentes actualizado o eliminado, y nuevos registros. Cualquier aplicación con los permisos adecuados puede agregar, quitar o actualizar los datos desde cualquier otra aplicación incluyendo bases de datos de los nativos de Android [33].

Muchas bases de datos nativas están disponibles como proveedores de contenido, accesible por aplicaciones de terceros, incluida la gestión de contactos del teléfono, guardar los medios de comunicación, y otras bases de datos nativas.

### **4.4.8 Acceso al Hardware mediante Android**

Las capacidades multimedia de los dispositivos móviles son una consideración importante para muchos consumidores y desarrolladores. La plataforma abierta Android y la filosofía independiente del proveedor asegura que ofrece una API de multimedia capaz de reproducir y grabar una amplia gama de imagen, audio, vídeo y formatos, tanto local y en directo [33], [34].

La API de la cámara y la plataforma multimedia OpenCORE exponen estas capacidades a las aplicaciones, proporcionando una funcionalidad multimedia completa.

#### **Reproducción de audio y video**

Android incluye un completo reproductor de medios para simplificar la reproducción de audio y video. Android 2.1 (API a nivel 7) soporta los siguientes formatos multimedia para la reproducción como parte de la estructura base. Teniendo en cuenta que algunos dispositivos pueden admitir la reproducción de formatos de archivo adicionales [33], a continuación en la Tabla 4.9 se puede observar los formatos de audio y video que admite Android 2.1.

**Tabla. 4.9. Formatos de Audio y Video para Android 2.1.**

Audio	Video
AAC LC/LTP	H.263
HE-AACv1 (AAC+)	H.264 AVC
HE-AACv2 (Enhanced AAC+)	MPEG-4 SP
AMR-NB, AMR-WB	
MP3, MIDI	

## Reproductor de medios

La reproducción multimedia de Android está a cargo de la clase Media Player. Puede reproducir archivos multimedia almacenados en los recursos de aplicaciones, archivos locales, proveedores de contenido, o transmitido desde una dirección URL de la red. En cada caso, el formato de archivo y el tipo de multimedia que se está reproduciendo se abstrae dependiendo del requerimiento del desarrollador.

Las transiciones a través de la máquina de estado se pueden describir de la siguiente forma:

- Iniciar el reproductor de medios con los medios para reproducir.
- Preparar el Media Player para la reproducción.
- Iniciar la reproducción.
- Pausa o detener la reproducción antes de su finalización.
- La reproducción completa.

Para reproducir un recurso de los medios de comunicación se necesita crear una instancia de Media Player nuevo, iniciar con una fuente de medios de comunicación, y prepararlo para la reproducción [33].

En cada caso, una vez que se haya terminado la reproducción, `release` es el objeto de Media Player para liberar los recursos asociados.

Android soporta un número limitado de objetos simultáneamente Media Player, Android puede hacer excepciones en tiempo de ejecución cuando el sistema se queda sin recursos [33].

### **Preparando Audio para la reproducción**

Hay un número de maneras en que puede reproducir contenido de audio a través del Media Player. Se puede incluir como un recurso de aplicación, reproducirlos de archivos locales ó proveedores de contenidos, o desde un mando a distancia URL [33].

### **Paquetes de audio como un recurso de aplicación**

Se puede incluir archivos de audio del paquete de solicitud mediante la inclusión en la carpeta `res/raw` de los recursos de jerarquía. Los recursos de `raw` no se comprimen o manipulan de ninguna forma cuando se ingresa en la aplicación, las convierte en una forma ideal para almacenar archivos comprimidos antes como contenido de audio [33].

Para acceder a `raw` sólo se tiene que utilizar el nombre de archivo sin una extensión en minúsculas.

### **Inicialización de contenido de audio para la reproducción**

Para reproducir contenido de audio con el reproductor de medios de comunicación, es necesario crear un nuevo objeto de Media Player y establecer el origen de datos del audio en cuestión. Para reproducir audio con el reproductor multimedia, se puede utilizar el método `create` [33].

## Grabación de audio y video.

La grabación multimedia está a cargo de la clase `MediaRecorder`. Se puede usar para grabar archivos de audio y/o video que se puede utilizar en las aplicaciones, o añadidos a la tienda de los medios de comunicación. Para grabar audio o video, se crea un nuevo objeto `Media Recorder` [33].

```
MediaRecorderMediaRecorder = new MediaRecorder ();
```

Antes de poder grabar cualquier medio de comunicación en Android, la aplicación necesita permisos de `RECORD_AUDIO` y `RECORD_VIDEO` de la siguiente manera.

```
<uses-permissionandroid:name="android.permission.RECORD_AUDIO"/>  
<uses-permissionandroid:name="android.permission.RECORD_VIDEO"/>
```

El grabador de medios de comunicación permite especificar la fuente de audio y vídeo, el formato de archivo de salida, y los codificadores de audio y video para usarse cuando se graba el archivo.

Al igual que el `Media Player`, el grabador de medios de comunicación gestiona la grabación como una máquina de estados. Esto significa que el orden en el que configura y administrar la grabadora de los medios de comunicación es importante [33]. En términos más simples, las transiciones a través de la máquina de estado se pueden describir como sigue:

- Creación de un nuevo grabador de medios de comunicación.
- Asignar las fuentes de entrada que se desea grabar.
- Definir el formato de salida.
- Especificar el audio y el codificador de vídeo, `framerate`, y el tamaño de salida.
- Seleccionar un archivo de salida.
- Preparar para la grabación.
- Registro.
- Grabación final.

## Configuración y control de grabación de vídeo

Antes de la grabación se debe especificar las fuentes de entrada, el formato de salida, el audio y el codificador de vídeo y un archivo de salida en ese orden [33].

- Los métodos `setAudioSource` y `setVideoSource` permiten especificar una constante `MediaRecorder.AudioSource` o `MediaRecorder.VideoSource` estática que define la fuente de audio y vídeo, respectivamente.
- Una vez que se haya seleccionado las fuentes de entrada, se selecciona el formato de salida usando el método `setOutputFormat` para especificar un `MediaRecorder.OutputFormat` constante.
- Se utiliza el ajuste `[Audio/Video] Encoder`, métodos para especificar un archivo de audio o video encoder constante de la `MediaRecorder`. `[Audio/Video]` clase `Encoder`.
- Por último, se asigna un archivo para almacenar los archivos grabados utilizando el método `setOutputFile`. Cuando se haya terminado, `callstop` pone fin a la reproducción, seguido por la liberación de la grabadora de los medios de comunicación de los recursos.

## Utilización de la cámara

La popularidad de las cámaras digitales sobre todo en terminales de telefonía ha provocado su caída en los precios al igual que su tamaño se ha reducido drásticamente. Hoy es difícil de encontrar un teléfono móvil sin una cámara, y los dispositivos Android no son una excepción [33].

## Utilizando Intents para tomar fotos

La forma más fácil de tomar una foto utilizando la cámara del dispositivo es utilizando los medios de comunicación `ACTION_IMAGE_CAPTURE`. Esto abrirá la actividad de la cámara, permitiendo modificar la configuración de la imagen de forma manual, y que le impide tener que reescribir la aplicación de toda la cámara [33].

La acción de captura de imágenes es compatible con dos modos, en miniatura y la imagen completa.

- **Miniatura:** De manera predeterminada, la imagen tomada por la acción de captura de imagen devuelve un mapa de bits en miniatura de los datos adicionales dentro del parámetro de los Intents que retorna un `onActivityResult`.
- **Imagen completa:** Si se especifica un URI de salida utilizando un `MediaStore.EXTRA_OUTPUT` adicional en el intento de lanzamiento, la imagen a tamaño completo tomadas por la cámara se guardará en la ubicación especificada.

Una vez que haya tomado la foto, se puede agregar a la tienda de los medios de comunicación, o a un proceso para su uso dentro de la aplicación antes de eliminarla.

### **El control de la cámara y tomar fotos**

Para acceder al hardware de la cámara directamente, es necesario agregar el permiso de Cámara en la aplicación Manifest [33]. Se utiliza la clase `Camera` para ajustar la configuración de la cámara, especificar las preferencias de la imagen, y tomar fotos. Para acceder a los servicios de la cámara, se utiliza el método estático abierto sobre la clase `Camera`.

### **Control y seguimiento de ajustes de la cámara y opciones de imagen**

La configuración de la cámara se almacena utilizando un objeto `Camera.Parameters`, se accede llamando al método `getParameters` en el objeto de la cámara. Con el fin de modificar la configuración de la cámara, se utiliza los métodos establecidos en el objeto antes de llamar a los parámetros del método `Camera.Parameters` y pasar al objeto ya modificado. Android 2.0 (nivel de API 5) presentó una amplia gama de parámetros de la cámara, cada uno con un setter y getter [33], se puede ver en la siguiente Tabla 4.10 los principales parámetros de la cámara.

**Tabla. 4.10. Parámetros de la Cámara.**

Parámetros de la Cámara	Función
[get / set] SceneMode	Toma o devuelve una SCENE_MODE_ * cadena estática constante de la clase de cámara Parameters. Cada modo de escena describe un tipo en particular (fiesta, playa, puesta de sol, etc.)
[get / set] FlashMode	Toma o devuelve una FLASH_MODE_ * cadena estática constante. Permite especificar el modo de flash, como encendido, apagado, reducción de ojos rojos, o el modo de linterna.
[get / set] BalanceBlancos	BalanceBlancos Toma o devuelve una cadena estática WHITE_BALANCE_ * constante para describir el balance de blancos de la escena a fotografiar.
[get / set] ColorEffect	Toma o devuelve una cadena estática EFFECT_ * constante para modificar la forma de presentar la imagen. Efectos de color disponibles son los tonos: sepia o blancos y negros.
[get / set] FocusMode	Toma o devuelve una cadena estática FOCUS_MODE_ * constante para especificar cómo la cámara autofocus debe tratar de enfocar la cámara.

Las Actividades de la Cámara también se pueden utilizar para leer o especificar el tamaño, la calidad y los parámetros de formato para la imagen, miniatura y vista previa de la cámara [33]. La Tabla 4.11 explica cómo configurar algunos de estos valores.

**Tabla. 4.11. Actividades de la Cámara.**

Actividades	Forma de configurar
JPEG y la calidad de imagen	Utilizar los métodos y setJpegQuality, setJpegThumbnailQuality, pasando de un valor entero entre 0 y 100, donde 100 es la mejor calidad.
Imagen, vista previa, y el tamaño de las miniaturas	Se usasetPictureSize, setPreviewSize, setJpegThumbnailSize para especificar una altura y anchura de la imagen, vista previa, y la miniatura, respectivamente.
Imagen y vista previa de píxeles	Se usasetPictureFormat, setPreviewFormat para establecer el formato de imagen mediante una constante estática de la clase PixelFormat.
Vista previa	Se usa setPreviewFrameRate para especificar la velocidad de cuadro de vista previa en fps (frames por segundo).

Cada dispositivo potencialmente puede apoyar un subconjunto diferente de valores de estos parámetros. La clase `Camera parameters` también incluye una serie de métodos `getSupported` para encontrar opciones válidas para mostrar al usuario, o confirmar un valor del parámetro deseado [33].

### **Monitoreo de enfoque automático**

Si la Cámara admite el enfoque automático, y está habilitada, se puede monitorear el éxito de la operación de enfoque automático, añadiendo un `AutoFocusCallback` con el objeto de la cámara [33].

El controlador de eventos `onAutoFocus` recibe un parámetro cuando el estado de la cámara de enfoque automático ha cambiado, y un parámetro booleano que indica si el éxito del enfoque automático se ha logrado.

### **Usando la vista previa de la cámara**

El acceso a streaming de vídeo de la cámara significa que se puede incorporar vídeo en directo a sus aplicaciones. Algunas de las aplicaciones para Android más interesantes utilizan esta funcionalidad como base para la aplicación de realidad aumentada (el proceso de superposición dinámica de datos contextuales, tales como los detalles de hitos o puntos de interés en la parte superior de una cámara en vivo de alimentación).

Al igual que las clases `Media Player` y `Media Recorder`, la previa de la cámara se muestra en un `SurfaceHolder`. Para ver la secuencia de cámara en vivo dentro de su aplicación, se debe incluir una visión de la superficie dentro de la interfaz de usuario, además implementar un `SurfaceHolder.Callback` para escuchar la construcción de una superficie válida, antes de pasar al método `setPreviewDisplay` del objeto `Camera` [33].

Una llamada a `startPreview` comenzará la transmisión y `stopPreview` va a terminar. También se puede asignar un `PreviewCallback` por cada cuadro de vista

previa, lo que permite manipular o mostrar cada cuadro de vista previa de forma individual.

### Tomar una foto

Se toma una fotografía llamando al método `takePicture` en un objeto de la cámara, este pasa por un método `ShutterCallback` y dos implementaciones `PictureCallback` (uno para el formato RAW y JPEG para una codificación de imágenes).

Cada imagen devuelve una matriz de bytes que representa la imagen en el formato adecuado, mientras que la devolución de llamada del obturador se dispara inmediatamente después de que el obturador se cierra [33].

### Reconocimiento de Voz

Desde Android 1.5 (nivel de API 3), Android ha apoyado la entrada de voz y reconocimiento de voz utilizando la clase `RecognizerIntent`. Este API permite aceptar la entrada de voz en su aplicación utilizando el cuadro de diálogo de entrada de voz estándar como se muestra en la Figura 4.27.



**Figura. 4.27. Reconocedor de Voz de Android.**

El Reconocimiento de voz se inicia llamando a `startNewActivityForResult`, y pasando un `Intent` especificando el `RecognizerIntent.ACTION_RECOGNIZE_SPEECH` que es una acción constante [33]. El `Intent` incluye `RecognizerIntent.EXTRA_LANGUAGE_MODEL` para especificar el modelo de lenguaje utilizado

para analizar el audio de entrada ó `LANGUAGE_MODEL_FREE_FORM` y `LANGUAGE_MODEL_WEB_SEARCH`, ambos están disponibles como constantes estáticas de la clase `RecognizerIntent` [33].

También se puede especificar una serie de comandos opcionales para controlar el idioma, contar con resultados potenciales, y la pantalla del sistema mediante las constantes siguientes vistas en la Tabla 4.12.

**Tabla. 4.12. Comandos opcionales de control de audio.**

Comandos	Función
<code>EXTRA_PROMPT</code>	Especifica una cadena que se mostrará en la ventana de entrada de voz para pedir al usuario hablar.
<code>EXTRA_MAXRESULTS</code>	Especifica un valor entero para limitar el número de resultados posibles de reconocimiento.
<code>EXTRA_LANGUAGE</code>	Especifica un lenguaje constante de la clase <code>Locale</code> para especificar un idioma de entrada que no sea el dispositivo por defecto. Se puede encontrar el valor predeterminado actual llamando al método estático <code>getDefault</code> en la clase <code>Locale</code> .

El motor que maneja el reconocimiento de voz puede no ser capaz de entender de entrada todos los idiomas disponibles en la clase `Locale`. No todos los dispositivos que incluyen soportan reconocimiento de voz [33]. En tales casos, generalmente es posible descargar la librería de reconocimiento de voz del `Android Market`.

Cuando el usuario ha terminado la entrada de su voz, el sonido resultante será analizado y procesado por el motor de reconocimiento de voz. Los resultados serán devueltos a través del controlador `onActivityResult` como una lista de arreglos adicionales de cuerdas en el `EXTRA_RESULTS` [33].

## **CAPITULO V**

### **DISEÑO DEL SOFTWARE DE RECONOCIMIENTO DE LA DENOMINACIÓN DE DÓLARES AMERICANOS**

#### **5.1 INTRODUCCIÓN**

En este capítulo se detallaran los conceptos y procedimientos utilizados para el diseño del software de reconocimiento de denominación de dólares americanos.

Dichos conceptos abarcan teoría de reconocimiento de patrones, herramientas de procesamiento digital de imágenes - PDI y análisis de componentes principales - PCA, siendo estos temas la base fundamental sobre la cual se desarrollará el software.

En la parte final se expondrá la forma en la cual todos estos conceptos se conjugan para obtener el diseño del software que será implementado en el teléfono móvil inteligente con Android SO.

#### **5.2 RECONOCIMIENTO DE PATRONES**

El Reconocimiento de patrones es la base sobre la cual se desarrollará el software de reconocimiento de dólares americanos ya que plantea la estructura que debe tener el software para poder funcionar eficientemente

##### **5.2.1 Concepto de Reconocimiento de Patrones.**

El reconocimiento de patrones – PCA llamado también lectura de patrones, identificación de figuras y reconocimiento de formas, consiste en el reconocimiento de

patrones de señales, estos patrones se obtienen a partir de los procesos de segmentación, extracción de características y descripción dónde cada objeto queda representado por una colección de descriptores. El sistema de reconocimiento debe asignar a cada objeto su categoría o clase [35] (conjunto de entidades que comparten alguna característica que las diferencia del resto).

Para poder reconocer los patrones se siguen los siguientes procesos:

- Adquisición de datos.
- Extracción de características.
- Toma de decisiones.

### 5.2.2 Enfoques del Reconocimiento de Patrones

Los principales enfoques del Reconocimiento de Patrones son los siguientes [35]:

- **Enfoque estadístico-probabilístico:** Su principal clasificador está basado en la teoría de la probabilidad, y específicamente en el teorema de Bayes<sup>16</sup>. Es históricamente el primer enfoque que existió y probablemente el más desarrollado.
- **Enfoque sintáctico-estructural:** Se basa en la teoría de autómatas y lenguajes formales para hacer la clasificación. Se enfoca más en la estructura de las cosas a clasificar, que en mediciones numéricas.
- **Enfoque lógico-combinatorio:** Creado en la antigua Unión Soviética, tiene como principal característica el poder trabajar con variables de todo tipo, aunque sus algoritmos suelen ser de complejidad alta.
- **Enfoque neuronal:** Se basa en modelos matemáticos de las neuronas biológicas, o dicho de otra forma, trata de emular la forma en cómo interactúan nuestras neuronas.
- **Enfoque Asociativo:** Creado en el Centro de Investigación en Computación del IPN en 2002, utiliza los modelos de memorias asociativas para crear clasificadores robustos.

---

<sup>16</sup>**Teorema de Bayes:** Vincula la probabilidad de A dado B con la probabilidad de B dado A.

### 5.2.3 Sistema Básico de Reconocimiento

Un sistema completo de reconocimiento de patrones incluye un sensor, que recoja fielmente los elementos del universo a ser clasificado, un mecanismo de extracción de características cuyo propósito es extraer la información útil, eliminando la información redundante e irrelevante, y finalmente una etapa de toma de decisiones o clasificador en la cual se asigna a la categoría apropiada los patrones de clase desconocida a priori como se muestra en la Figura 5.1 [35].

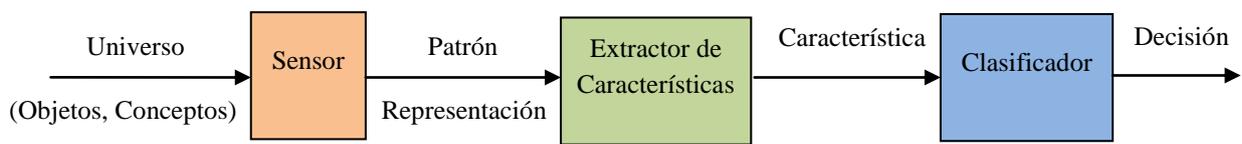


Figura. 5.1. Sistema Básico de Reconocimiento.

#### Sensor

El sensor es el dispositivo encargado de la adquisición de datos y el que tiene la capacidad de transformar magnitudes físicas o químicas, llamadas variables de instrumentación, en magnitudes eléctricas. Las variables de instrumentación dependen del tipo de sensor y pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, etc.

#### Sistema de Extracción de Características

Es el proceso de generar características que puedan ser usadas en el proceso de clasificación de los datos. En ocasiones viene precedido por un pre procesamiento de la señal, necesario para corregir posibles deficiencias en los datos debido a errores del sensor, o bien para preparar los datos de cara a posteriores procesos en las etapas de extracción de características o clasificación [35].

Las características elementales están explícitamente presentes en los datos adquiridos y pueden ser pasados directamente a la etapa de clasificación. Las

características de alto orden son derivadas de las elementales y son generadas por manipulaciones o transformaciones en los datos.

### Sistema de Clasificación

La clasificación trata de asignar las diferentes partes del vector de características a grupos o clases, basándose en las características extraídas. En esta etapa se usa lo que se conoce como aprendizaje automático, cuyo objetivo es desarrollar técnicas que permita al sistema de reconocimiento aprender [35].

Según tengamos constancia o no de un conjunto previo que permita al sistema aprender, la clasificación puede ser supervisada, parcialmente supervisada o no supervisada.

- **Clasificación supervisada:** También es conocida como clasificación con aprendizaje. Se basa en la disponibilidad de áreas de entrenamiento. Se trata de áreas de las que se conoce a priori la clase a la que pertenecen y que servirán para generar una signatura espectral característica de cada una de las clases. Se denominan clases informacionales en contraposición a las clases espectrales que genera la clasificación no supervisada [35], [36].

Algunos métodos de la clasificación supervisada son:

- **Funciones discriminantes:** Si son dos clases, se busca obtener una función  $g(x)$  tal que para un nuevo objeto, *si*  $g(O) \geq 0$  se asigna a la clase 1 y en otro caso a la 2. Si son múltiples clases se busca un conjunto de funciones y el nuevo objeto se ubica en la clase donde la función tome el mayor valor.
- **Vecino más cercano:** Un nuevo objeto se ubica en la clase donde esté el objeto de la muestra original que más se le parece.
- **Redes neuronales artificiales:** Denominadas habitualmente RNA o en sus siglas en inglés ANN. Imitan a las redes neuronales reales en el desarrollo de tareas de aprendizaje.

- **Clasificación parcialmente supervisada:** También conocida como de aprendizaje parcial. En estos problemas existe una muestra de objetos sólo en algunas de las clases definidas.
- **Clasificación no supervisada:** También conocida como clasificación sin aprendizaje. Se utilizan algoritmos de clasificación automática multivariante en los que los individuos más próximos se van agrupando formando clases.
  - **Restringida:** El número de clases en la que se estructurará la muestra está previamente definido.
  - **Libre:** El número de clases en la que se estructurará la muestra, depende exclusivamente de los datos.

### 5.3 PROCESAMIENTO DIGITAL DE IMÁGENES

El procesamiento digital de imágenes - PDI es un conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar su calidad y facilitar la extracción de información, dichas técnicas contribuirán a mejorarla eficacia y eficiencia del sistema de reconocimiento de dólares americanos, a continuación se presentan algunos conceptos relacionados con el PDI [36].

#### 5.3.1 Imagen Digital

Una imagen en escala de grises puede ser definida como una función bidimensional  $f(x, y)$ <sup>17</sup>, donde  $x$  e  $y$  son coordenadas espaciales, y el valor de  $f$  en un par de coordenadas dado es denominado intensidad o nivel de gris. Cuando  $x$ ,  $y$  y  $f$  son cantidades discretas, la imagen es llamada imagen digital. De este modo, ésta puede ser representada como una matriz de  $M \times N$  elementos llamados píxeles [36].

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix} \quad (5.1)$$

<sup>17</sup>  $f(x, y)$ : Una imagen digital puede ser representada como función de dos variables  $x$  e  $y$

### 5.3.2 Modelos de Color

El uso del color en el procesamiento de imágenes es de importancia ya que suele simplificar la identificación y extracción de objetos en una escena. Un modelo de color especifica un sistema de coordenadas y un sub-espacio en el cual un color es definido por un punto. Muchos modelos están orientados al hardware como lo son el RGB (rojo, verde y azul), CMY (cian, magenta y amarillo) y el CMYK (cian, magenta, amarillo y negro); otros están orientados hacia aplicaciones donde se pretende manipular el color, tales como HSI (tono, saturación e intensidad), HSV (tono, saturación y valor) y el YCbCr (luminancia, crominancia azul y crominancia roja) [36].

#### Modelo de color RGB

Este modelo se basa sobre un sistema de coordenadas cartesianas y el sub-espacio en el cual se define el color es el cubo normalizado entre  $[0,1]$  como se muestra en la Figura 5.2 (a). Los valores rojo (R), verde (G) y azul (B) están sobre los vértices  $(1, 0, 0)$ ,  $(0, 1, 0)$  y  $(0, 0, 1)$  respectivamente; el origen corresponde al negro, en el punto más lejano a éste se ubica el blanco, y el resto de los vértices corresponden a los colores secundarios: cian, magenta y amarillo [36].

El resto de colores está compuesto por la combinación de los colores primarios R, G y B, y son puntos dentro del cubo definido por un vector extendido desde el origen.

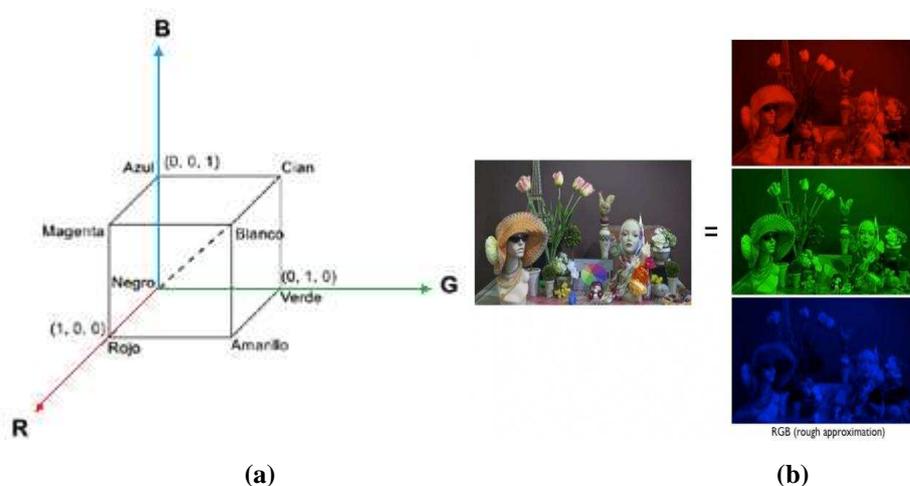


Figura. 5.2. Modelo de color RGB:

(a) Ubicación de un píxel en el plano RGB.      (b) Imagen en el modelo de color RGB.

Las imágenes representadas en el modelo RGB consisten de tres planos, uno por cada color primario como en la Figura 5.2 (b). El número de bits usados para representar cada píxel es llamado profundidad.

### Modelo de color YCbCr

Éste es un modelo de color usado en sistemas de video y fotografía digital, donde Y es la componente de luminancia y Cb y Cr son las componentes de crominancia azul y roja [36].

YCbCr es el equivalente digital al modelo YUV (Sistema de TV). Concentra la mayor parte de la información de la imagen en la luminancia y menos en la crominancia. El resultado es que los elementos de YCbCr están menos correlacionados y pueden ser codificados por separado [36].

En el modelo YCbCr, cada cuadro se compone de una luminosidad única (en blanco y negro) de la imagen, y dos componentes de crominancia. La información del Verde proviene de una combinación de los datos de crominancia y luminancia.

YCbCr fue creado con el fin de reducir el ancho de banda de la señal de vídeo mediante la eliminación de la luminancia redundante en la señal de RGB [36] (cada uno con su propio valor de luminancia).

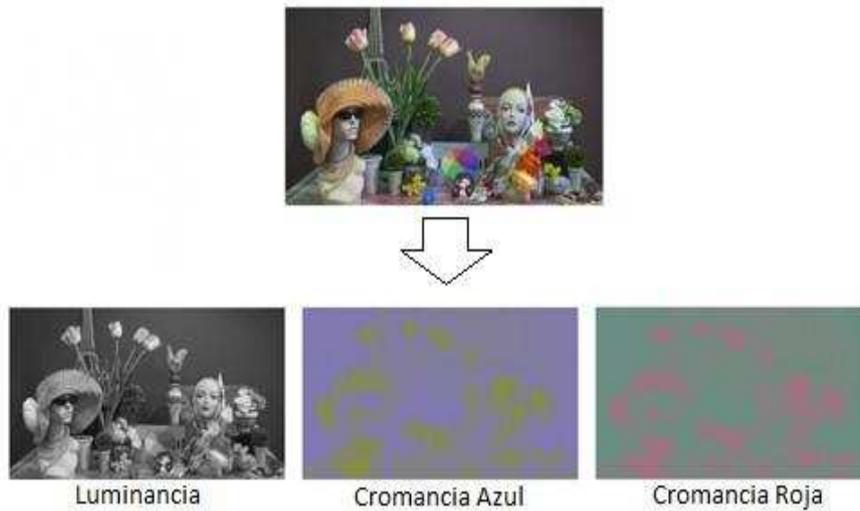
Esto también permite la reducción de la información de color sin afectar a la resolución real de la imagen. Este proceso se llama Sub muestreo de color.

El paso de un sistema RGB a un YCbCr es obtenido mediante las ecuaciones:

$$Y = 0,3R + 0,6G + 0,1B \quad (5.2)$$

$$Cb = \frac{B - Y}{2} + 0,5 \quad (5.3)$$

$$Cr = \frac{R - Y}{2} + 0,5 \quad (5.4)$$



**Figura. 5.3. Imagen representada en el modelo YCbCr.**

### Sub muestreo de color

El submuestreo de color es un método para reducir la resolución del color sin afectar la resolución general de la imagen. Mediante la reducción de la información del color, la velocidad de bits de la secuencia de vídeo también se puede reducir [36].

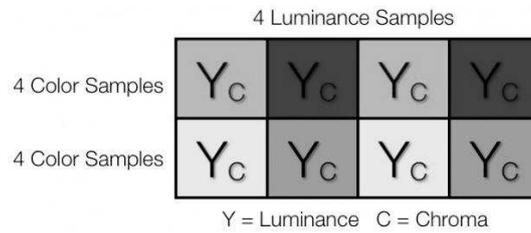
En el modelo de color YCbCr, esto se logra fácilmente ya que la parte de la luminancia de la imagen es independiente de la parte de la imagen Croma [36].

En la Figura 5.3 anterior se puede ver que la señal de luminancia es esencialmente una imagen en blanco y negro, los ojos humanos son mejores para ver el brillo, por lo que la señal Y se mantiene generalmente a resolución completa.

Sin embargo, si se reduce la resolución de las imágenes en color (Cb, Cr), nos costara distinguir la diferencia.

A se puede ver los números relacionados con la compresión como '4:2:2' o '4:2:0'. Esto se refiere al método y la cantidad de sub muestreo de color.

Estos números se pueden deducir de manera sencilla, si se tiene un rectángulo de 4 píxeles de ancho por 2 píxeles de altura como en la Figura 5.4.



**Figura. 5.4. Imagen de 4:4:4.**

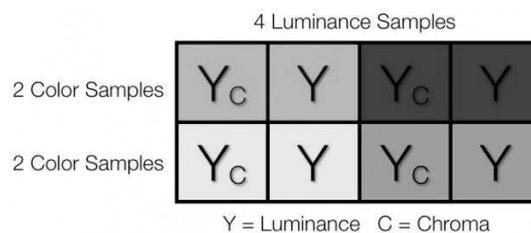
El primer valor de muestreo representa el muestreo de luminancia (Y) en la imagen a través de ambas líneas. Como se puede ver en la Figura 5.4, se tiene valores de Y para los cuatro píxeles en ambas líneas [36]. Por lo tanto, tienen una resolución de luminancia completa.

El segundo valor de muestreo representa la cantidad de información de croma en la primera línea de la caja. En la Figura 5.4, se tiene cuatro muestreos de crominancia (C), así que se tiene una resolución completa de croma en esa línea.

El tercer valor de muestreo representa la cantidad de información de croma en la segunda línea de la caja. En la Figura 5.4, se tiene cuatro muestreos de crominancia (C), así que tenemos una resolución completa de croma en esa línea.

Por lo tanto, la imagen en la Figura 5.4 se podría llamar 4:4:4, la información tanto de color como de luminancia en la señal están completas. Cuando se empieza a reducir esa información, las cosas se ponen un poco más complejas.

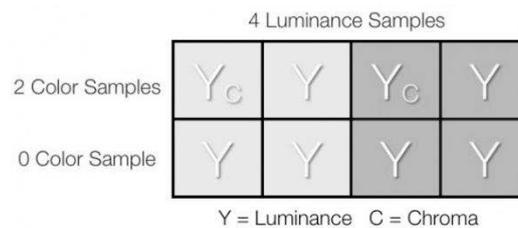
Para el caso de 4:2:2:



**Figura. 5.5. Imagen de 4:2:2.**

Como se puede ver en la Figura 5.5, todavía se tiene la información de luminancia completa (el primer 4 de los 4:4:4), pero se tiene menos información de croma, dos en la primera línea y dos en la segunda. Así que efectivamente se ha reducido la resolución de color a la mitad en ambas líneas [36]. La información de muestreo de croma que ha sido eliminada, ha sido promediada dentro de los 2 valores para mantener la imagen exacta.

Para el caso de 4:2:0:



**Figura. 5.6. Imagen de 4:2:0.**

La información de luminancia se mantiene todavía en la resolución completa, pero se tiene dos muestreos de croma en la primera línea y ninguno en la segunda. Los muestreos de croma son un promedio de los dos píxeles horizontales y de los dos píxeles verticales.

### Profundidad de color

La profundidad de color describe la cantidad de información almacenada en cada píxel de datos. A medida que aumenta la profundidad de bits, también aumenta el número de colores que se pueden representar.

En el caso de una imagen RGB de 24 bits, cada píxel tiene 8 bits de datos por color (RGB), así que para cada canal de color del píxel tiene 2 a la 8va potencia = 256 posibles variaciones color. En el caso de una imagen de 10-bit RGB, cada canal de color tendría 2 a la 10ma potencia = 1024 variaciones de color [36].

### 5.3.3 Histograma de una Imagen

El histograma de una imagen es la representación gráfica de la distribución que existe de las distintas tonalidades de grises con relación al número de píxeles o porcentaje de los mismos. La representación de un histograma ideal sería la de una recta horizontal, ya que eso nos indicaría que todos los posibles valores de grises están distribuidos de manera uniforme en la imagen.

El histograma de una imagen digital con niveles de gris en el rango  $[0, L - 1]$  es una función discreta  $P(r) = n_r$  donde  $r$  representa un nivel de gris y  $n_r$  representa el número de píxeles que tienen ese valor de gris [36].

Es muy frecuente normalizar un histograma dividiendo cada uno de sus valores por el número total de píxeles en la imagen, denotado por  $n$ . De tal modo que el histograma nos quedaría  $P(r_k) = n_k/n$ , para  $k = 0, 1, \dots, L - 1$ . De esta manera,  $P(r_k)$  da una estimación de la probabilidad de la ocurrencia del nivel de gris  $r_k$ . Hay que notar que la suma de todos los componentes de un histograma normalizado es igual a 1. En la Figura 5.7 se puede observar los diferentes ejemplos de tipos de histograma [36].

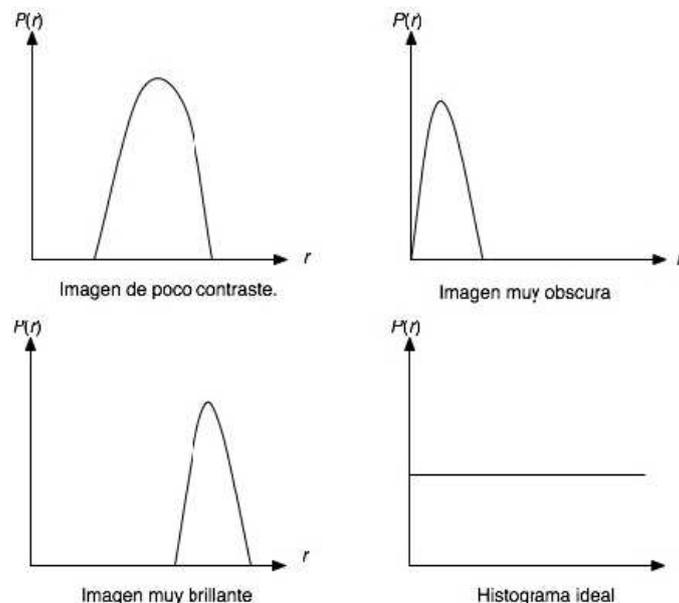


Figura. 5.7. Tipos de Histograma.

## Manipulación del Histograma

El histograma es de gran utilidad en el tratamiento de imágenes ya que con él se puede variar el aspecto de las imágenes. Los objetivos de la manipulación del histograma son:

- Mejora en la calidad de la imagen con técnicas de realzado, que tratan de eliminar efectos no deseados tales como sombras y reflejos a la vez que aumentan el contraste.
- Fijar umbrales en procesos de binarización para la extracción de regiones en las imágenes.

Hay tres técnicas básicas de transformación del histograma, que resultan muy útiles para modificar el brillo y el contraste de las imágenes originales. Estas son: contracción, expansión y desplazamiento.

### Contracción del Histograma

Esta técnica produce una disminución en el contraste del histograma como se puede observar en la Figura 5.8. Dicha técnica está definida por:

$$g(i, j) = \left[ \frac{C_{MAX} - C_{MIN}}{f(i, j)_{MAX} - f(i, j)_{MIN}} \right] [f(i, j) - f(i, j)_{MIN}] + C_{MIN}, \quad (5.5)$$

Donde  $g(i, j)$  el nivel de gris de la imagen de salida,  $C_{MAX}$  y  $C_{MIN}$  es el máximo y mínimo valor respectivamente deseado en la compresión del histograma,  $f(i, j)_{MAX}$  y  $f(i, j)_{MIN}$  son el máximo y mínimo nivel de gris respectivamente de la imagen de entrada [36].

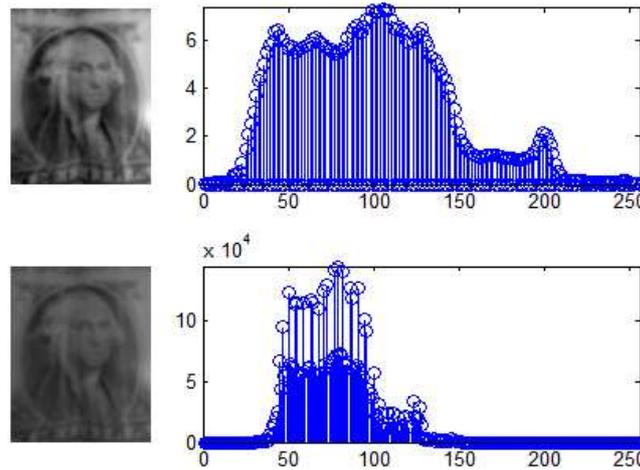


Figura. 5.8. Compresión del Histograma.

### Expansión del Histograma

Ésta operación opuesta a la contracción del histograma. Su función es:

$$g(i, j) = \left[ \frac{f(i, j) - f(i, j)_{\text{MIN}}}{f(i, j)_{\text{MAX}} - f(i, j)_{\text{MIN}}} \right] [\text{MAX} - \text{MIN}] + \text{MIN}, \quad (5.6)$$

Donde  $f(i, j)_{\text{MAX}}$  y  $f(i, j)_{\text{MIN}}$  son el máximo y mínimo nivel de gris respectivamente de la imagen de entrada, MAX y MIN son el máximo y mínimo nivel de gris respectivamente que se desea que tenga la imagen de salida.

Esta ecuación toma una imagen de entrada  $f(x, y)$  y expande el histograma a lo largo del rango de valores completo de los niveles de gris. Esto tiene el efecto de incrementar el contraste de una imagen de bajo contraste. Si se desea que la expansión no cubra el rango total posible de niveles de gris, se pueden especificar diferentes valores para MAX y MIN [36].

Si muchos valores de una imagen caen dentro de un pequeño rango y existe un cierto número de valores extremos, entonces el histograma abarca todo el rango de valores y una expansión pura del histograma no mejora la imagen como se mira en la Figura 5.9. En este caso suele ser una práctica habitual recortar los niveles de gris en los

extremos a los valores de gris más bajo y más alto del rango (para una imagen de 8 bits serían 0 y 255) [36].

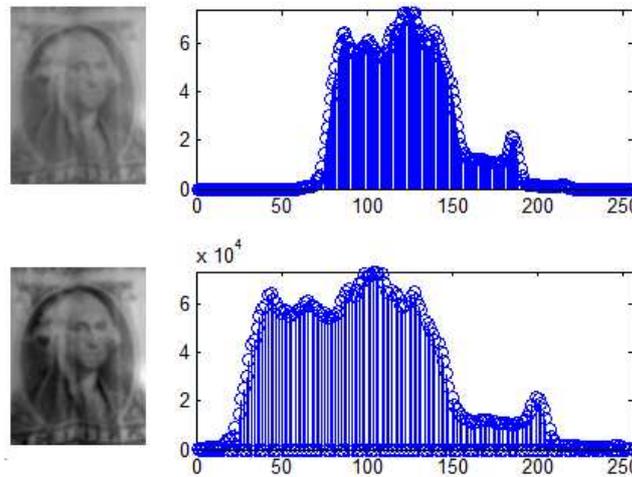


Figura. 5.9. Expansión del Histograma.

### Desplazamiento del Histograma

Se usa para aclarar u oscurecer una imagen, pero manteniendo la relación entre los valores de los niveles de gris como se observa en la Figura 5.10. Esta operación puede llevarse a cabo por la simple adición o sustracción de un número fijo (DES en la siguiente fórmula) a todos los valores del nivel de gris:

$$g(i, j) = f(i, j) + \text{DES}, \quad (5.7)$$

los valores que sobrepasen el máximo y el mínimo por exceso, se redondean, respectivamente, al máximo y mínimo posibles permitidos. Un valor DES positivo incrementa el brillo de la imagen y, por tanto, la aclara, mientras que un valor negativo la oscurece al disminuir el brillo [36].

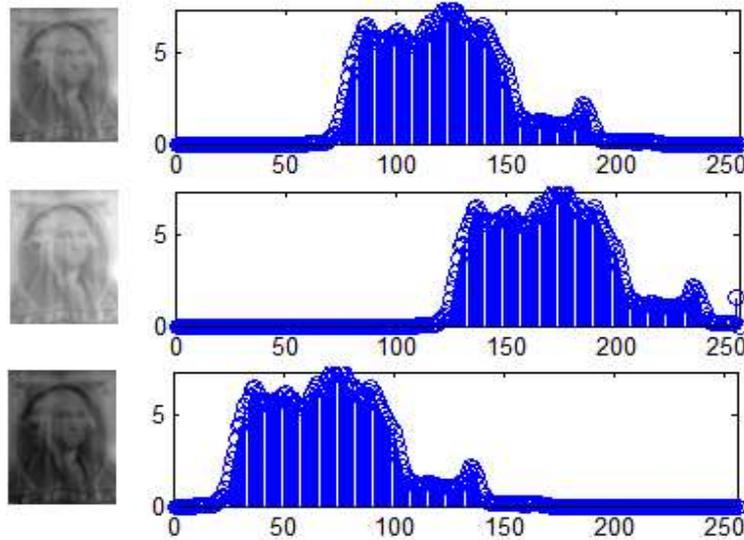


Figura. 5.10. Desplazamiento del Histograma.

### 5.3.4 Submuestreo de imágenes

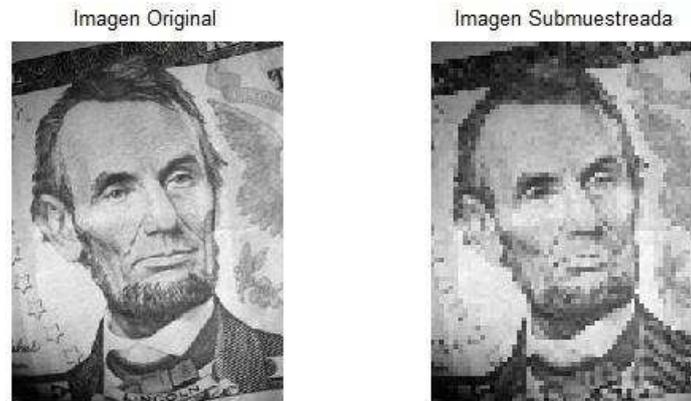
Realizar ciertos cálculos con la imagen en su resolución original es muchas veces muy costoso. Una alternativa más eficiente, es submuestreando la imagen. Submuestreo significa generar una imagen a partir de tomar muestras periódicas de la imagen original, de tal forma que esta quede más pequeña. Si se considera la imagen  $I(m, n)$  definida como:

$$I(m, n) = \begin{bmatrix} V_{11} & V_{12} & V_{13} & \cdots & V_{1n} \\ V_{21} & V_{22} & V_{23} & \cdots & V_{2n} \\ V_{31} & V_{32} & V_{33} & \cdots & V_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ V_{m1} & V_{m2} & V_{m3} & \cdots & V_{mn} \end{bmatrix} \quad (5.8)$$

A la cual se desea sub muestrear para obtener la mitad de su tamaño original, así la nueva imagen quedaría compuesta solo por elementos cuyos índices sean impares con lo cual la imagen  $I(m, n)$  quedaría definida como  $I_s(m, n)$  [36].

$$I_s(m, n) = \begin{bmatrix} V_{11} & V_{13} & V_{15} & \cdots & V_{1(n-2)} \\ V_{31} & V_{33} & V_{35} & \cdots & V_{3(n-2)} \\ V_{51} & V_{53} & V_{55} & \cdots & V_{5(n-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ V_{(m-2)1} & V_{(m-2)3} & V_{(m-2)5} & \cdots & V_{(m-2)(n-2)} \end{bmatrix} \quad (5.9)$$

En la Figura 5.11 se puede observar este método de submuestreo en la imagen de un billete de 5 dólares.

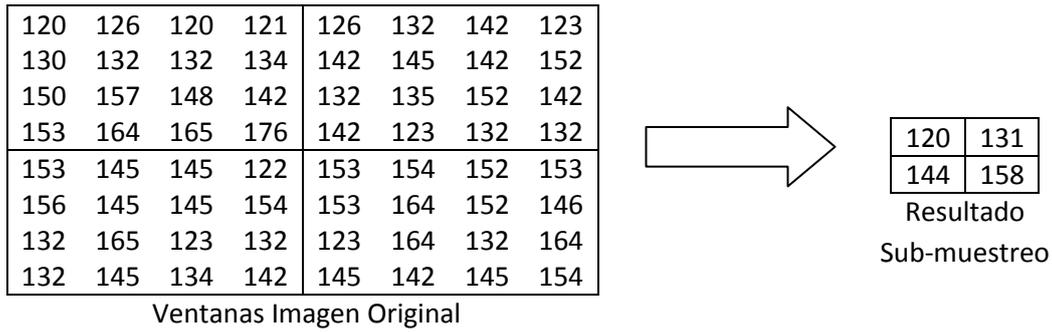


**Figura. 5.11. Submuestreo de Imágenes.**

El método de submuestreo anteriormente mencionado es poco eficiente ya que como se puede observar en la Figura 5.11 inserta ruido y disminuye en exceso los detalles de la imagen al tener cambios abruptos en los niveles de gris entre pixeles vecinos, por lo cual para el presente trabajo se consideró dos tipos de sub muestreo que no insertan ruido y conservan los detalles suavizando los bordes de la imagen, dichos métodos de submuestreos se explican a continuación.

### **Sub muestreo por media y por mediana**

Estos procesos de submuestreo consideran un ventaneado de la imagen original para submuestrearla, el número de ventanas dependerá del número de pixeles de la imagen que se desea obtener con el submuestreo ya que cada ventana se utilizará para obtener el valor de cada pixel de la imagen procesada, en la Figura 5.12 se muestra la forma en la que cada ventana de la imagen original aporta con un valor de pixel de la imagen resultante.



**Figura. 5.12. Ventaneado de Imagen.**

El submuestreo por media utiliza la media de los valores de la ventana para obtener el valor del pixel de la imagen que se desea obtener [36].

120	126	120	121
130	132	132	134
150	157	148	142
153	164	165	176

$$pix = \frac{120 + 126 + 120 + \dots + 176}{16} = 141$$

En el submuestreo por mediana, se obtiene el valor central o mediana de todos los valores en la ventana para obtener el valor del pixel [36].

120	126	120	121
130	132	132	134
150	157	148	142
153	164	165	176

$$pix = MED(120 \ 126 \ 120 \ \dots \ 176) = 142$$

En la Figura 5.13 se puede observar los dos tipos de submuestreo con la misma imagen.



Figura. 5.13. Comparación entre submuestreo por media y por mediana.

Con ambos tipo de submuestreo se obtiene una imagen que conservan las principales características sin inserción de ruido, lo cual contribuirá para que el proceso de extracción de características sea más rápido ya que se posee menos información por procesar.

### 5.3.5 Técnicas de Filtrado

Se trata de métodos para resaltar o suprimir, de forma selectiva, información contenida en una imagen a diferentes escalas espaciales, para destacar algunos elementos de la imagen, o también para ocultar valores anómalos.

El proceso de filtrado consiste en la aplicación a cada uno de los pixeles de la imagen de una matriz de filtrado de tamaño  $N \times N$  (generalmente de  $3 \times 3$  como en la Figura 5.14 que corresponde a una máscara de filtrado por media) compuesta por números enteros y que genera un nuevo valor mediante una función del valor original y los de los pixeles circundantes. El resultado final se divide entre un escalar, generalmente la suma de los coeficientes de ponderación. Los filtros se pueden expresar mediante una ecuación (5.10):

$$ND'_{i,j} = \frac{ND_{i-1,j-1} + ND_{i,j-1} + ND_{i+1,j-1} + ND_{i-1,j} + ND_{i,j} + ND_{i+1,j} + ND_{i-1,j+1} + ND_{i,j+1} + ND_{i+1,j+1}}{9} \quad (5.10)$$

donde  $i$  y  $j$  representan la fila y la columna de cada pixel,  $ND_{i,j}$  su nivel digital y  $ND'_{i,j}$  el nivel digital obtenido tras hacer el filtrado [3].

1	1	1
1	1	1
1	1	1

 $v=9$ 

**Figura. 5.14. Mascara de Filtrado por media.**

Mediante diferentes combinaciones de parámetros asignados a los diferentes pixeles circundantes se pueden conseguir diferentes efectos. En general los efectos deseados se relacionan con el aspecto borroso que tienen las imágenes tomadas con la cámara del dispositivo móvil, debido a las diferentes condiciones de luz.

Hay que tener en cuenta que los bordes de la imagen no podrán procesarse, ya que la ventana de filtrado saldría fuera de la imagen. De este modo se pierden  $N - 2$  filas y  $N - 2$  columnas por cada lado de la imagen [36].

Los filtros más utilizados son los de paso bajo (suavizan la imagen), de paso alto (aumentan el contraste), los filtros direccionales (detectan en la imagen estructuras que siguen una determinada dirección) y los de detección de bordes (permiten identificar y aislar objetos con propiedades homogéneas dentro de la imagen) [36]. En el presente trabajo se utilizará un filtro pasa bajos de media ponderada el cual será explicado a continuación:

### **Filtros de paso bajo**

Su objetivo es suavizar la imagen, son útiles cuando se supone que la imagen tiene gran cantidad de ruido y se quiere eliminar. También pueden utilizarse para resaltar la información correspondiente a una determinada escala (tamaño de la matriz de filtrado); por ejemplo en el caso de que se quiera eliminar la variabilidad asociada a los tipos de cubierta presentes en la imagen uniformizando de esta manera su respuesta. Existen varias posibilidades:

- **Filtro de la media:** Asigna al pixel central la media de todos los pixeles incluidos en la ventana. La matriz de filtrado estaría compuesta por unos y el

divisor sería el número total de elementos en la matriz, la Figura 5.15 muestra la máscara del filtro de la media.

- **Filtro de media ponderada:** Los elementos de la matriz de filtrado no son todos 1 sino que se da más peso a uno de ellos (generalmente el central) para obtener un resultado más parecido a la imagen original y evitar que aparezca borrosa [37].
- **Filtro de la mediana:** Tiene la ventaja de que el valor final del pixel es un valor real presente en la imagen y no un promedio, de este modo se reduce el efecto borroso que tienen las imágenes que han sufrido un filtro de media. Además el filtro de la mediana es menos sensible a valores extremos. El inconveniente es que resulta más complejo de calcular ya que hay que ordenar los diferentes valores que aparecen en los pixeles incluidos en la ventana y determinar cuál es el valor central [37].

### Filtros de paso alto

Su objetivo es realzar las zonas de mayor variabilidad eliminando la componente media, precisamente los que detectan los filtros paso bajo. Por otra parte la respuesta de cada pixel está contaminada por la de los pixeles vecinos ya que, considerando la superficie del billete como Lambertiana<sup>18</sup>, la radiación reflejada por un pixel se reparte hacia los pixeles vecinos [37]. Los filtros de paso alto consiguen también eliminar en parte esta contaminación, existen varios métodos:

- **Filtro Laplaciano:** El filtro Laplaciano es una medida 2-D isotrópica de la 2<sup>da</sup> derivada espacial de una imagen. El Laplaciano de una imagen destaca las regiones donde hay cambios bruscos de intensidad y por tanto se suele utilizar para detección de bordes. El Laplaciano se aplica frecuentemente a una imagen que previamente ha sido suavizada mediante un filtro paso bajo de suavizado, con el fin de reducir su sensibilidad al ruido [38].

---

<sup>18</sup>Superficie Lambertiana: Es aquella que refleja la energía uniformemente hacia todas las direcciones.

- **Filtro Menos Laplaciano:** es un filtro que realza rasgos lineales en las imágenes, restando la imagen original de la imagen obtenida con el Laplaciano utilizando la siguiente mascara [38].

0	-1	0
-1	5	-1
0	-1	0

Div=1Div

**Figura. 5.15. Filtro menos Laplaciano.**

## 5.4 ANÁLISIS DE COMPONENTES PRINCIPALES

El Análisis de las Componentes Principales - PCA, del inglés Principal Component Analysis es una técnica estadística que realiza una transformación lineal y ortogonal de los datos a un nuevo sistema de coordenadas tal que la máxima varianza queda proyectada sobre la primer coordenada, la segunda mayor varianza sobre la segunda coordenada, y así sucesivamente. Transforma un conjunto de variables correlacionadas en un conjunto de variables no correlacionadas, y simplifica la transformación encontrando las componentes más cercanas a las variables originales pero ordenadas en forma decreciente al orden de su varianza. Puede ser utilizada para reducir la dimensionalidad del conjunto de datos reteniendo aquellas características del conjunto que mayor contribución hacen a su varianza, manteniendo las primeras componentes principales de más bajo orden. Tales componentes frecuentemente contienen la mayor información del conjunto de datos [39].

El PCA fue desarrollado principalmente para la detección e identificación de rostros utilizando imágenes faciales, sin embargo al ser dicha técnica muy eficiente en el reconocimiento de patrones e involucrar poca carga computacional se ha considerado como la alternativa más viable para ser implementada en el sistema reconocedor de la denominación de dólares americanos [39].

EL funcionamiento del PCA al utilizarse en identificación de rostros consiste en extraer las características más relevantes, las cuales pueden o no estar relacionadas a las características faciales como boca, nariz, ojos, etc. En términos matemáticos, se quiere

encontrar las componentes principales de la distribución de caras, o lo que es lo mismo, los eigenvectors de la matriz de covarianza del conjunto de caras [39].

Estos eigenvectors pueden ser vistos como el conjunto de vectores que caracterizan la variación entre las imágenes faciales. A partir de esto, cada cara puede ser representada exactamente como una combinación lineal de los eigenvectors. Además, cada cara se puede aproximar usando aquellos que tienen asociados los eigenvalues más altos, es decir, los que responden a la mayor varianza del conjunto de imágenes faciales.

Al utilizarse el PCA en el sistema reconocedor de la denominación de dólares americanos su funcionamiento será el mismo que con imágenes faciales, ya que las imágenes de dólares americanos al igual que las faciales guardan similitud entre sí.

#### 5.4.1 Calculo de Eigenfaces

Sea  $I(x, y)$  una imagen facial o de dólares americanos en escala de grises de  $N \times N$ . Dicha imagen puede ser representada como un vector de dimensión  $N^2$  o equivalentemente, un punto en el espacio  $N^2$  – dimensional, por lo tanto, un conjunto de imágenes es mapeado a una colección de puntos en un espacio inmenso, considerando el tamaño de las imágenes con las que se suele trabajar. De esta manera, un conjunto de imágenes, sean estas faciales o de dólares americanos, dado que todas tienen un aspecto similar, se mapean en un sub espacio del espacio de imágenes y así pueden ser descritas por un sub espacio de menor dimensión. La idea es encontrar los vectores que mejor describan la distribución de caras dentro del espacio completo de imágenes. Estos vectores de longitud  $N^2$ , llamados eigenfaces, definen el sub espacio de imágenes, el cual se denomina espacio de caras [39].

Sea  $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M$  un conjunto de imágenes faciales o de dólares americanos. La cara media o meanface  $\Psi$  del conjunto es definida por:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n. \quad (5.11)$$

Cada cara difiere de ésta por el vector  $\Phi_i = \Gamma_i - \Psi$ . A partir de este conjunto de vectores, se busca un conjunto de  $M$  vectores ortonormales,  $u_n$ , los cuales mejor describan los datos. El  $k$ -ésimo vector,  $u_k$ , es seleccionado tal que

$$\lambda_k = \frac{1}{M} \sum_{n=1}^M (u_k^T \Phi_n)^2 \quad (5.12)$$

Es un máximo, sujeto a que los vectores  $u_k$  sean ortonormales. Los vectores  $u_k$  y los escalares  $\lambda_k$  son los eigenvectors y eigenvalues, respectivamente, de la matriz de covarianza:

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T \quad (5.13)$$

$$C = AA^T,$$

donde  $A = [\Phi_1 \Phi_2 \Phi_3 \dots \Phi_M]$ . La matriz  $C$  resultante es de  $N^2 \times N^2$  y obtener  $N^2$  eigenvectores y eigenvalores es una tarea costosa para el tamaño de imágenes comúnmente procesadas [39].

Una consideración a tener en cuenta es que si el número de puntos en el espacio de imágenes es menor que la dimensión del mismo  $M < N^2$ , habrá solamente  $M - 1$  eigenvectores significantes, el resto tendrá asociados eigenvalues iguales a cero. En este caso, se tienen  $M$  imágenes caras de dimensión  $N^2$  y se pueden encontrar los eigenvectors  $N^2$  - dimensionales resolviendo los eigenvectors para una matriz de  $M \times M$  y luego realizar una combinación lineal de las imágenes faciales  $\Phi_l$  [39].

Consideremos los eigenvectors  $v_l$  de  $A^T A$  tal que

$$A^T A v_l = \mu_l v_l. \quad (5.14)$$

Pre-multiplicando ambos lados por  $A$ , se tiene

$$AA^T A v_l = \mu_l A v_l. \quad (5.15)$$

Reemplazando la ecuación (5.7) en (5.9)

$$CAv_l = \mu_l Av_l. \quad (5.16)$$

De aquí se ve que  $Av_l$  son los eigenvectors de la matriz de covarianza  $C$ . Entonces, se construye la matriz de  $M \times M$ ,  $L = A^T A$ , donde  $L_{mn} = \Phi_m^T \Phi_n$ , y se encuentran los  $M$  eigenvectores  $v_l$ , de  $L$ . Estos vectores determinan la combinación lineal de las  $M$  imágenes faciales para obtener los eigenvectors [39].

$$u_l = Av_l. \quad (5.17)$$

Es importante normalizar  $u_l$  tal que  $\|u_l\| = 1$  esto se lo puede hacer de la siguiente manera

$$u_l = Av_l D_l^{-1/2} \quad (5.18)$$

Donde  $D_l$  es una matriz diagonal con los eigenvalues de  $L$ .

De esta forma se reducen enormemente los cálculos del orden del número de píxeles en las imágenes ( $N^2$ ) al orden del número de imágenes en el conjunto ( $M$ ). Los eigenvalues asociados permiten ordenar los eigenvectors de acuerdo al aporte que hacen a la variación entre imágenes [39].

#### 5.4.2 Proyección y reconstrucción de una imagen utilizando eigenfaces

Una imagen facial o de Dólares Americanos  $\Gamma$  es proyectada sobre el espacio de caras por medio de la operación:

$$w_k = u_k^T (\Gamma - \Psi), \quad \text{para } k = 1, \dots, M. \quad (5.19)$$

Los pesos  $w_k$  forman un vector  $\Omega^T = [w_1 w_2 w_3 \dots w_M]$  que describen el aporte de cada eigenface en representar la imagen cara [39], tratando a las eigenfaces como un conjunto base para las imágenes faciales o de dólares americanos.

La reconstrucción exacta de una cara se realiza a través de una combinación lineal de las eigenfaces ponderadas con los respectivos pesos  $w_k$  [39].

$$\Gamma = \sum_{k=1}^M w_k u_k. \quad (5.20)$$

### 5.4.3 Reconocer una cara

Un método para determinar si una determinada cara se encuentra en la base de entrenamiento, consiste en encontrar la clase de cara  $k$  que minimice la distancia Euclideana<sup>19</sup>.

$$\varepsilon_k = |(\Omega - \Omega_k)| \quad (5.21)$$

$$\varepsilon_k < \text{umbral}$$

En donde  $\Omega_k$  es un vector que describe la  $k$  –ésima clase de cara. Si  $\varepsilon_k$  es menor que un “umbral predefinido”, entonces, la cara se clasifica como perteneciente a la clase  $k$ . Lo cual implica, haber identificado la cara en la base de datos correspondiente [39].

## 5.5 DISEÑO DEL SOFTWARE DE RECONOCIMIENTO DE DÓLARES AMERICANOS PARA NO VIDENTES

### 5.5.1 Análisis de requerimientos

El software de reconocimiento de dólares americanos debe cumplir con los siguientes requerimientos:

#### Requerimientos funcionales

- El sistema debe ser capaz de capturar una imagen a través de la cámara del teléfono móvil inteligente con Android SO.
- Identificar la denominación del dólar americano.

<sup>19</sup> Distancia Euclideana: Es la distancia ordinaria entre dos puntos de un espacio Euclídeo, Índice cuantitativo que mide la separación existente entre dos unidades de observación según los valores que ellas posean en un conjunto de variables.

- El sistema debe ser optimizado para el uso de personas con discapacidad visual a través de notificaciones audibles y táctiles.

### Requerimientos no funcionales

- El sistema debe funcionar bajo condiciones de luz variantes (luz natural y artificial).
- El sistema debe ejecutarse en tiempo real por lo cual las operaciones y procesos computacionales deben ejecutarse en un tiempo aceptable.

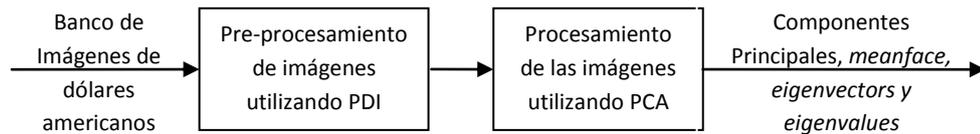
### 5.5.2 Diseño de la aplicación

Para el diseño de la aplicación se tomara en cuenta los requerimientos funcionales y no funcionales, para lo cual se aplicaran los conceptos descritos en las secciones anteriores de este capítulo en cada uno de los diferentes bloques de cada etapa de la aplicación.

Para la determinación de las diferentes etapas de la aplicación se consideró los conceptos de reconocimiento de patrones, con lo cual se definió que el sistema deberá tener dos etapas, las cuales son:

- **Etapas de aprendizaje:** En esta etapa se realiza el aprendizaje del sistema reconocedor de la denominación de dólares americanos, utilizando para esto un banco de imágenes de dólares americanos, las cuales serán pre-procesadas utilizando técnicas y herramientas de procesamiento digital de imágenes - PDI para mejorar su calidad y desechar información redundante, a continuación se procesaran para extraer los componentes principales o eigenvectores del banco de imágenes, las cuales serán almacenados para ser utilizados en la etapa de reconocimiento como se muestra en la Figura 5.16.
- **Etapas de reconocimiento:** En esta etapa, las imágenes recogidas a través de la cámara del dispositivo móvil, serán reconocidas por sistema de reconocimiento de la denominación de dólares americanos, para lo cual dichas imágenes pasaran

por un bloque de pre-procesamiento, al igual que en la etapa de aprendizaje, para mejorar su calidad y posteriormente someterlas al proceso de extraer sus componentes principales con la ayuda de los eigenvectors obtenidos en la etapa de aprendizaje por último las imágenes serán reconocidas gracias a sus componentes principales, el esquema se puede observar en la Figura 5.17.



**Figura. 5.16. Bloques de la Etapa de Aprendizaje.**



**Figura. 5.17. Bloques de la Etapa de Reconocimiento.**

## **CAPITULO VI**

# **DESARROLLO E IMPLEMENTACION DEL SOFTWARE DE RECONOCIMIENTO DE LA DENOMIAACIÓN DE DOLARES AMERICANOS**

### **6.1 INTRODUCCIÓN**

En este capítulo se detallara el desarrollo e implementación del software de Reconocimiento de la Denominación de Dólares Americanos dirigido a personas con discapacidad visual, para lo cual se considerará los conceptos descritos en los anteriores capítulos.

### **6.2 DESARROLLO DEL SOFTWARE DE RECONOCIMIENTO DE LA DENOMINACIÓN DE DÓLARES AMERICANOS**

Siguiendo el diseño del software del capítulo 5 que nos plantea dos etapas importantes como es la Etapa de Aprendizaje y la Etapa de Reconocimiento, plantearemos el desarrollo de estas etapas.

#### **6.2.1 Desarrollo de la Etapa de Aprendizaje**

Para el desarrollo de esta etapa se utilizará Matlab que un software desarrollado por MathWorks<sup>TM</sup> especializado en el tratamiento de matrices, por ende es ideal para el procesamiento digital de imágenes, ya que como se menciona en el capítulo 5 una imagen digital se considera un arreglo bidimensional o matriz.

La función de esta herramienta dentro del presente proyecto radica en la generación de los componentes principales eigenvectors, obtenidos del banco de imágenes de dólares americanos, como se puede observar en la Figura 5.17.

### Obtención del Banco de Imágenes de dólares Americanos

Debido a que se utiliza el algoritmo de PCA, se tiene que tomar varias muestras de los billetes para almacenarlas en un Banco de Imágenes de Dólares Americanos y utilizarlas en el reconocimiento. El algoritmo de PCA considerará la imagen central de los billetes que posee el rostro de los presidentes norteamericanos en su parte frontal como se puede ver en la Figura 6.1 y en el anexo A3 se puede observar la totalidad de este banco de imágenes.



Figura. 6.1. Obtención del Banco de Imágenes de Dólares Americanos.

Para la conformación del Banco de Imágenes se tomaran 24 muestras distribuidas de la siguiente manera vistas en la Tabla 6.1.

Tabla. 6.1. Obtención del Banco de Imágenes de dólares Americanos

Cara	No. De Muestras	Ángulos de Inclinación				Detalles
		0°	90°	180°	270°	
Frontal	24	±1,	±1,	±1,	±1,	Cada muestra se tomó con cierto grado de inclinación debido a que las personas no videntes no siempre podrán colocar los billetes en una posición exacta.
		±5,	±5,	±5,	±5,	
		±10	±10	±10	±10	

## Pre-Procesamiento de las imágenes del Banco de Imágenes utilizando Matlab

Una vez obtenidas las muestras se procederá a realizar un proceso de sub muestreo por media como el menciona en la sección 5.3.4, del capítulo anterior, ya que trabajar con la resolución original resultaría costoso computacionalmente.

A continuación las imágenes serán pasadas por un filtro de media ponderada el cual suavizara las imágenes evitando que parezca borrosa, cómo se muestra en la Figura 6.2.

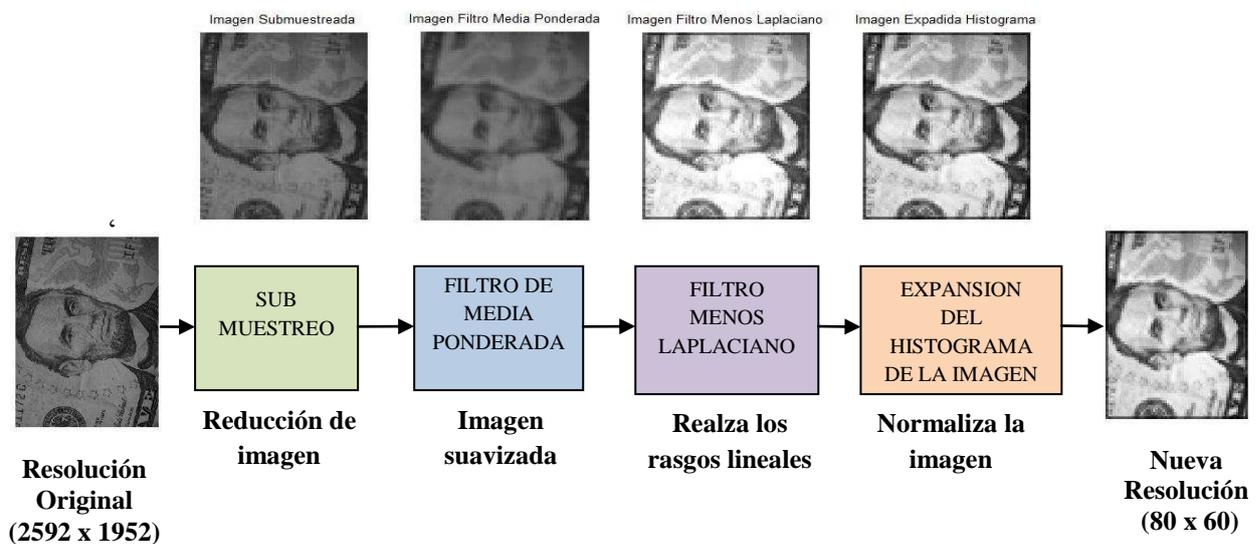
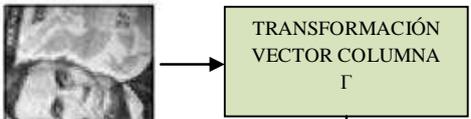
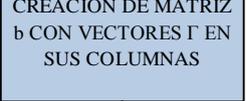
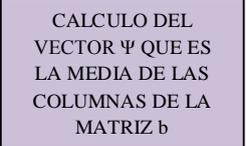
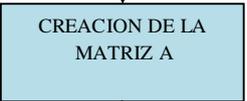
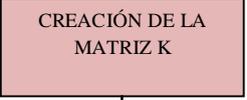
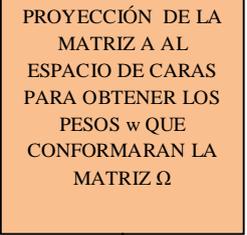
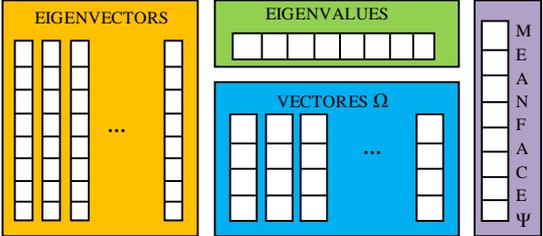


Figura. 6.2. Pre procesamiento de Imagen de muestra.

## Obtención de eigenvector sutilizando Matlab

Después de haber pre-procesado las imágenes para aumentar su contraste y disminuir su resolución se procederá a obtener los eigenvectors y engenvalues siguiendo el proceso descrito en el Capítulo 5, sección 5.4.1, con lo cual el procedimiento sería el descrito en la Tabla 6.2.

Tabla. 6.2. Proceso de obtención de Componentes Principales

DIAGRAMA DE FLUJO	COMANDOS MATLAB	DESCRIPCIÓN
	$\Gamma=I(:);$	El algoritmo de PCA requiere que las imágenes sean presentadas en forma de vector columna
	$b=[\Gamma_1 \Gamma_2 \Gamma_3 \dots \Gamma_m];$	Los vectores columna correspondientes a las imágenes serán ubicados dentro de las columnas de una matriz denominada b.
	$\Psi=\text{mean}(b,2);$	Se obtendrá un vector que contenga los valores medios el cual será denominado de todas las columnas de la matriz b.
	$[m\ n]=\text{size}(b);$ $A=b-(\text{diag}(\Psi)*\text{ones}(m, n));$	Se restará a todas las columnas de la matriz b el vector Ψ para obtener una matriz denominada A.
	$K=A' * A;$	Se obtendrá la matriz de covarianza de A'.
	$[VL, DL]= \text{eigs}(K, 12);$ $UL= A*VL*DL^{(-1/2)};$	Se calculará los eigenvectors correspondientes a K para posteriormente con ellos calcular los eigenvector de A.
	$O=VL' * a;$	Se calculará los pesos de las imágenes perteneciente al banco de imágenes
	Almacenamiento de los vectores y matrices provenientes del proceso en archivos ".xml" que pueda leer Android.	

## Almacenamiento de eigenvectors

El almacenamiento de los eigenvectors obtenidos a partir del procedimiento descrito en la Figura 6.3 se lo realizara en archivos (.xml), aprovechando la capacidad que nos brinda Matlab para crear y editar archivos de texto, de tal forma que la aplicación de Android, sea capaz de leer los datos como un recurso de la aplicación a través de la clase R.java como se describe en el capítulo 4, sección 4.3.5.

Dichos archivos serán guardados en la carpeta “/res/values/” de la aplicación de Android ya que responden a un tipo de recurso llamado arrays.xml que sirven para definir arreglos de vectores de tipo String o enteros (int) como se muestra en la Tabla 4.3.

### 6.2.2 Desarrollo de la etapa de Reconocimiento

Para el desarrollo de esta etapa se utilizará la herramienta Eclipse IDE (previamente explicado en detalle en el capítulo 4) y el lenguaje de programación Java el cual es el oficialmente soportado para el desarrollo de aplicaciones en Android, a la vez que para crear la aplicación se considerará el modelo descrito en la Figura 5.17.

### Obtención de una imagen utilizando la cámara de un dispositivo móvil inteligente con Android SO

Para la obtención de la imagen por medio de la cámara del teléfono móvil se deberá utilizar las clases expuestas en el capítulo 4, sección 4.4.8, en la cual se explica la forma para acceder al hardware del dispositivo.

Dicha obtención seguirá el siguiente procedimiento:

1. **Crear un objeto que herede de la clase SurfaceView e implemente la interface SurfaceHolder.Callback:** Mencionada en la sección nombrada anteriormente, para poder visualizar la secuencia de cámara en vivo dentro de la aplicación. Al implementar SurfaceHolder.Callback también se debe implementar sus métodos abstractos:

- **surfaceCreated:** Este método es inmediatamente llamado después de crear la superficie en la cual se visualizará los datos adquiridos por la cámara del teléfono móvil.
- **surfaceChanged:** Este método es llamado después de realizar cualquier cambio en la configuración de la cámara.
- **surfaceDestroyed.** Este método es llamado inmediatamente después que la superficie de visualización es destruida.

2. **Crear un objeto tipo Camera:** El cual proveerá todos los métodos necesarios para utilizar la cámara del dispositivo móvil, dicho objeto será global dentro de la clase que implemente `SurfaceHolder.Callback`.

Antes de poder configurar cualquier parámetro de la cámara o visualizar un frame, la cámara debe inicializarse, esto se lo hace a través de los métodos `open` del objeto `Camera`, dicha inicialización deberá ir dentro del método `surfaceCreated` ya que solo se deberá hacer una sola vez al iniciar la aplicación.

3. **Configurar los parámetros de la cámara:** Mediante la instanciación de un objeto de tipo `Camera.Parameters`, el cual proveerá los métodos descritos en la Tabla 4.10, dichos parámetros serán pasados al objeto tipo `Camera` a través del método `setParameters` del mismo objeto, la configuración de la cámara seguirá la descrita en la Tabla 6.3 :

**Tabla. 6.3. Configuración de los Parámetros de la Cámara.**

Método de la Clase <code>Camera.Parameters</code>	Configuración
<b>setSceneMode</b>	Llevará como parámetro <code>Camera.Parameters.SCENE_MODE_NIGHT</code> lo cual permitirá configurar la cámara en modo nocturno, lo que facilitará el reconocimiento en ambientes con poca luz.
<b>setFlashMode</b>	Llevará como parámetro el String <code>Camera.Parameters.FLASH_MODE_OFF</code> lo cual desactiva el flash de la cámara, ya que algunos dispositivos no lo poseen, además se utilizará la función de auto enfoque la cual no funciona si el flash esta activado.
<b>setPreviewFrameRate</b>	Cada parámetro lleva un entero el cual especifica la cantidad de frames por segundos que capturará la cámara, para el presente proyecto se lo configurará en 15 debido a que es un nivel óptimo para realizar el reconocimiento.

<b>setPreviewSize</b>	Como parámetros lleva el tamaño de ancho y alto que tendrá el frame que capturará la cámara, dichos valores se setearan en 320 para el ancho y 240 para el alto debido a que son los mínimos valores que se pueden setear.
<b>setFocusMode</b>	Llevará como parámetro el String <code>Camera.Parameters.FOCUS_MODE_AUTO</code> ya que esto permitirá utilizar la función de auto enfoque de la cámara.

A continuación, se indicará que se inicie la pre visualización de la cámara mediante el método `startPreview`.

El proceso anteriormente descrito se lo hará dentro del método `surfaceChanged`, debido a que la superficie de visualización cambia cada vez que lo hacen los parámetros de la cámara.

4. **Utilizar el método `setPreviewDisplay`:** Para iniciar la superficie sobre la cual se proyectará la pre-visualización de los frames capturados por la cámara, el presente paso es muy importante ya que sin una superficie inicializada no se podrá llamar al método `startPreview`. Este método deberá ser llamado dentro del método `surfaceCreated` ya que al igual que el método `open` solo debe ser llamado una sola vez al inicio de la aplicación.
5. **Capturar un frame de la cámara:** Utilizando el método `setPreviewCallback` se instancia un objeto de tipo `Camera.PreviewCallback`, el cual invocará a su método `onPreviewFrame` cada vez que un frame se visualice en la pantalla del teléfono móvil.

El método `onPreviewFrame` tiene entre sus parámetros un vector de datos tipo `byte` en el cual se almacenará el frame capturado por la cámara utilizando el modelo de color YUV o YCbCr con submuestreo de color 4:2:0 (previamente explicado en el capítulo 5, sección 5.3.2).

### Pre Procesamiento de la imagen capturada

Posteriormente de capturar la imagen se procederá a pre-procesarla, debido a que, al tratarse de una imagen obtenida con un dispositivo móvil la misma poseerá una

contaminación gracias a las diferentes condiciones de iluminación, así también, al ser uno de los requerimientos la identificación en tiempo real la imagen deberá ser sub-muestreada para reducir el tiempo de procesamiento.

Ya que el algoritmo del PCA solo requiere de una imagen en escala de grises, se procederá a trabajar solo con los datos de luminancia - Y provenientes de la cámara, dichos datos serán obtenidos a partir del procedimiento descrito en ítem anterior.

Los procesos por los cuales atravesará la imagen en escala de grises, serán los mismos por los cuales pasaron las imágenes pertenecientes al banco de imágenes de aprendizaje y mostrado en la Figura 6.2.

### **Proyección de la imagen al espacio de caras**

Una vez pre-procesada la imagen se procederá a proyectarla utilizando los eigenvalor y meanface obtenidos a partir del procedimiento descrito en la Tabla 6.2 y almacenados en la carpeta “/res/values/” de la aplicación, dichos vectores serán accedidos mediante la clase R.java (descrito en el Capítulo 4, sección 4.3.5) la cual es el enlace de la aplicación a todos sus recursos, en este caso los recursos serán vectores de tipo enteros.

La proyección de la imagen obtenida con la cámara del teléfono móvil se la realizará aplicando la Fórmula 5.19 a dicha misma en forma de vector columna, con este procedimiento se obtendrá los pesos  $w$  correspondientes a esta imagen, con lo cual se podrá realizar la comparación e identificación de la misma.

### **Comparación e Identificación de la Imagen capturada**

Para el procedimiento de comparación, se obtendrá la distancia Euclideana con la Fórmula 5.21 entre los pesos  $w$  de la imagen obtenida por la cámara y los pesos de las imágenes del banco de muestras obtenidos con el procedimiento descrito en la Tabla 6.2

El umbral descrito en la Fórmula 5.21 será obtenido a partir de pruebas en varios tipos de iluminación.

### **6.3 IMPLEMENTACION DEL SOFTWARE DE RECONOCIMIENTO DE LA DENOMINACIÓN DE DOLARES AMERICANOS**

El dispositivo que albergue la aplicación resultante de este proyecto deberá poseer como mínimo las siguientes características:

#### **6.3.1 Requerimientos de Hardware**

- Procesador mínimo de 540 MHz.
- Memoria RAM de 384 MB.
- Memoria ROM de 512 MB.
- Cámara de 3.2 megapíxeles (que soporte modo nocturno).
- Pantalla Táctil 16M colores TFT.
- Soporte para melodías MP3, WAV.

#### **6.3.2 Requerimientos de Software**

- Android 2.1 (Eclair).

#### **6.3.3 Instalación del programa en el dispositivo móvil inteligente con Android**

Para la instalación de la aplicación se lo puede hacer de dos formas:

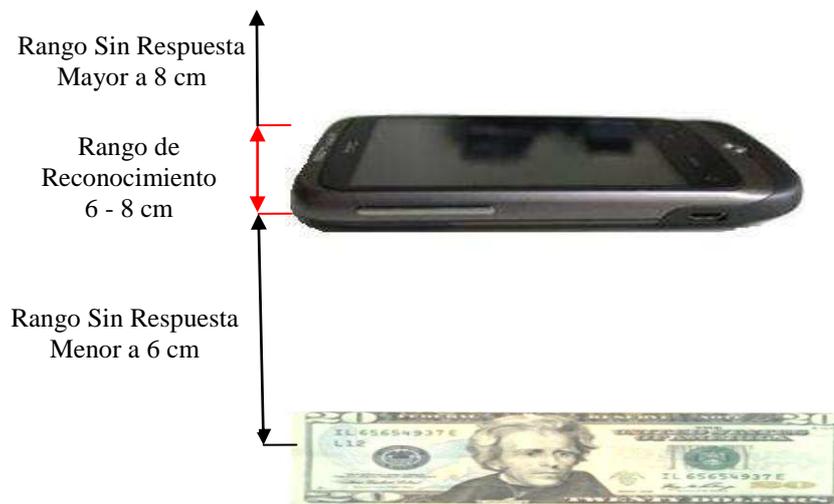
- Utilizando el modo de depuración del dispositivo móvil e instalándola directamente desde eclipse.
- Copiando al teléfono móvil el archivo (.apk) ubicado dentro de la carpeta “scr/” del proyecto de Eclipse para después acceder a él instalarlo manualmente desde el teléfono móvil.

## 6.4 PRUEBAS Y ANALISIS DE DESEMPEÑO

### 6.4.1 Pruebas de Reconocimiento

Para realizar las pruebas del software de reconocimiento obtenido, en primer lugar se debe saber la colocación del celular y el billete para una mayor efectividad (Seguir las instrucciones audibles del software).

Como se muestra en la Figura 6.3 el teléfono móvil inteligente debe estar colocado en forma horizontal encima del billete a una distancia en el rango de 6 a 8 cm que se la considera el rango de Reconocimiento óptimo.



**Figura. 6.3. Distancia entre el celular y el billete para el reconocimiento.**

Para saber en qué posición del billete obtenemos mayor efectividad de acierto se lleno la Tabla 6.4 y Tabla 6.5 para un teléfono con procesador de 540MHz y 600Mhz respectivamente, se tomo en ángulo de posición del billete Horizontal ( $0^\circ$  y  $180^\circ$ )y Vertical ( $90^\circ$  y  $270^\circ$ ), en una relación de 5 intentos de 10 segundos cada uno.

**Tabla. 6.4. Pruebas de Orientación del billete con un Procesador de 540 MHz.**

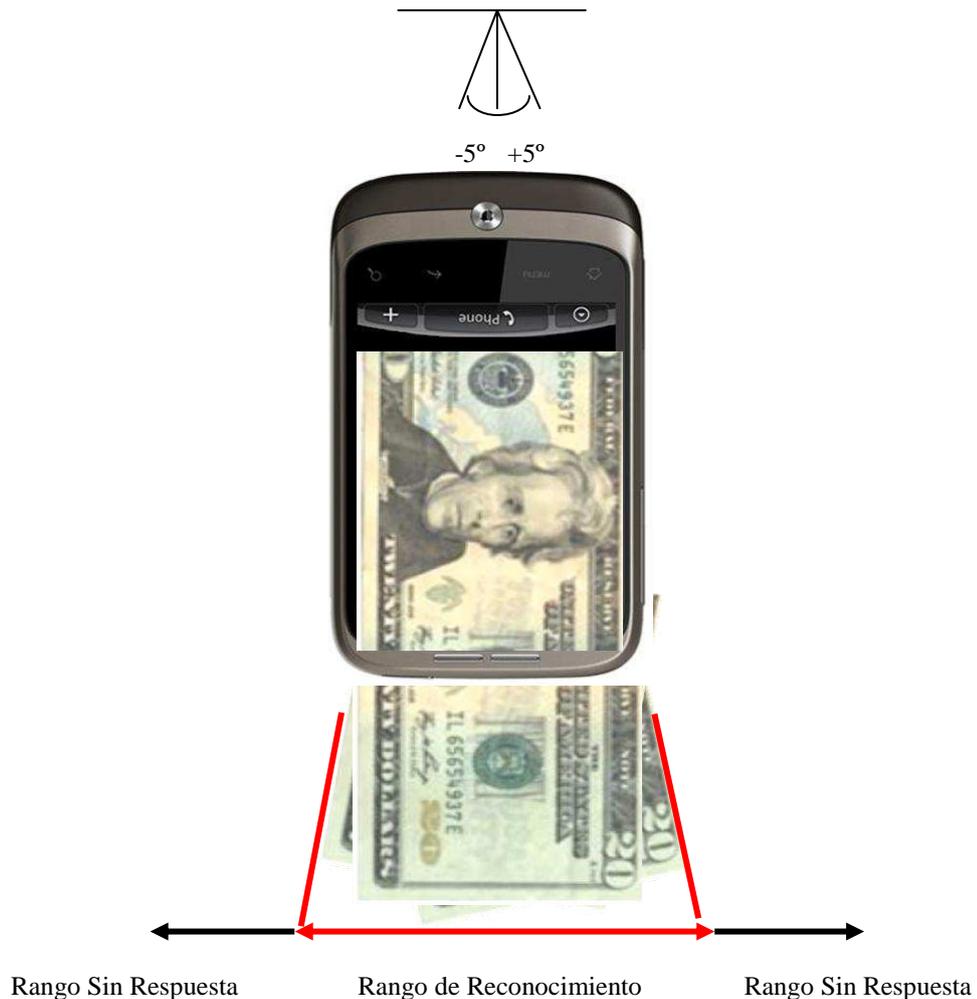
5 Intentos de 10 seg c/uno.																								
Resolución	1 Dólar						5 Dólares						10 Dólares						20 Dólares					
	Horizontal			Vertical			Horizontal			Vertical			Horizontal			Vertical			Horizontal			Vertical		
	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R
80x60	0	0	5	5	0	0	0	0	5	3	0	2	0	5	0	5	0	0	1	0	4	4	0	1
160x60	0	0	5	4	0	1	0	0	5	5	0	0	0	5	0	4	0	1	2	0	3	5	0	0
80x120	0	0	5	2	0	3	0	0	5	4	0	1	0	5	0	5	0	0	1	0	4	3	0	2
160x120	0	0	5	2	0	3	0	0	5	3	0	2	0	2	3	0	2	3	0	0	5	3	0	2

**Tabla. 6.5. Pruebas de Orientación del billete con un Procesador de 600 MHz.**

5 Intentos de 10 seg c/uno.																								
Resolución	1 Dólar						5 Dólares						10 Dólares						20 Dólares					
	Horizontal			Vertical			Horizontal			Vertical			Horizontal			Vertical			Horizontal			Vertical		
	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R	Aciertos	Errores	S/R
80x60	4	0	1	4	0	1	0	0	5	5	0	0	0	5	0	5	0	0	5	0	0	4	1	0
160x60	5	0	0	5	0	0	0	0	5	5	0	0	0	5	0	4	1	0	5	0	0	4	1	0
80x120	0	0	5	4	0	1	0	0	5	5	0	0	0	5	0	4	1	0	5	0	0	3	2	0
160x120	0	0	5	5	0	0	0	0	5	5	0	0	0	5	0	1	3	1	3	0	2	5	0	0

Los datos obtenidos de la Tabla 6.4 y Tabla 6.5, permite saber que existe mayor acierto cuando el billete se encuentra en posición vertical ( $90^\circ$  o  $270^\circ$ ), y existe mayor error y falta de respuesta en la posición horizontal ( $0^\circ$  y  $180^\circ$ ).

Además esta prueba permitió saber el rango de ángulos en el que el software reconoce óptimamente como se muestra en la Figura 6.4. El ángulo de posición del billete no debe ser mayor a  $\pm 5^\circ$  con respecto a la posición vertical del celular.



**Figura. 6.4. Ángulo de posición del billete con respecto al celular.**

En la Tabla 6.6 y Tabla 6.7 se muestra las 3 pruebas realizadas a cada billete con distintas resoluciones del banco de imágenes, además se hace una relación de números de aciertos en un tiempo de 10 segundos para obtener la velocidad de respuesta del software, posteriormente poder obtener la eficiencia del software en cuanto a reconocimiento y velocidad.

**Tabla. 6.6. Pruebas de Acierto en Procesador de 540 MHz.**

No. Aciertos en 10 seg.																		
Resolución	1 Dólar				5 Dólares				10 Dólares				20 Dólares				Prom Total	Efectividad
	1ra Prueba	2da Prueba	3ra Prueba	Prom	1ra Prueba	2da Prueba	3ra Prueba	Prom	1ra Prueba	2da Prueba	3ra Prueba	Prom	1ra Prueba	2da Prueba	3ra Prueba	Prom		
80x60	9	4	7	6,7	5	5	5	5,0	3	4	4	3,7	5	2	4	3,67	4,75	48%
160x60	1	7	6	4,7	6	5	6	5,7	4	4	4	4,0	5	5	5	5,00	4,83	48%
80x120	4	2	4	3,3	3	4	3	3,3	5	2	2	3,0	6	1	2	3,00	3,17	32%
160x120	0	0	2	0,7	2	1	1	1,3	0	1	0	0,3	1	1	1	1,00	0,83	8%

**Tabla. 6.7. Pruebas de Acierto en Procesador de 600 MHz.**

No. Aciertos en 10 seg.																		
Resolución	1 Dólar				5 Dólares				10 Dólares				20 Dólares				Prom Total	Efectividad
	1ra Prueba	2da Prueba	3ra Prueba	Prom	1ra Prueba	2da Prueba	3ra Prueba	Prom	1ra Prueba	2da Prueba	3ra Prueba	Prom	1ra Prueba	2da Prueba	3ra Prueba	Prom		
80x60	13	12	12	12,3	9	9	10	9,3	8	11	9	9,3	6	10	10	8,67	9,92	99%
160x60	10	13	11	11,3	9	11	9	9,7	10	8	6	8,0	11	9	9	9,67	9,67	97%
80x120	12	12	10	11,3	10	9	9	9,3	8	9	9	8,7	7	8	7	7,33	9,17	92%
160x120	9	9	11	9,7	7	8	7	7,3	7	9	8	8,0	5	6	4	5,00	7,50	75%

Según los datos que revelan la Tabla 6.6 y Tabla 6.7, el tamaño al cual se submuestra la imagen no es relevante en el dispositivo con el procesador de 600Mhz, sin embargo, si afecta en la lectura del dispositivo con procesador de 540MHz, a mas de esto se comprueba que la velocidad a la que el dispositivo da el resultado depende también de la duración de clip de sonido que describe el tipo de billete, ya que a si el dispositivo identifica el billete pero el clip de sonido no termina de reproducirse no podrá reproducir la el resultado de la nueva lectura.

También se pudo constatar que la resolución a la que el sistema tiene una mejor eficiencia es la de 160x60 ya que con esta resolución se obtiene una mayor cantidad de aciertos en el dispositivo de 540MHz y 600Mhz.

La velocidad de respuestas es de 2 a 3 segundos, que se considera que es una velocidad aceptable para el reconocimiento de la denominación de dólares americanos.

## **CAPITULO VII**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **7.1 CONCLUSIONES**

- El futuro de Linux y especialmente de Android es uno de los más claros y prometedores por sus propias características de inclusión en el desarrollo, seguridad y transparencia así como en la versatilidad de su modelo económico donde el Open Handset Alliance, emprendedores emergentes, universidades y empresas dedicadas al capital de riesgo están apostando a esta plataforma para ser pioneros en la nueva era de la información y las tecnologías libres.
- El Ecuador aun es un país que adopta tecnologías mas no las desarrolla en su totalidad, sin embargo con la investigación y el conocimiento adecuado se puede empezar a desarrollar aplicaciones para dispositivos móviles en Ecuador, como es el caso de este proyecto de desarrollo de una aplicación para dispositivos celulares con Android SO.
- El algoritmo de Análisis de Componentes Principales – PCA es muy adecuado para el reconocimiento de patrones, sin embargo, depende mucho de las condiciones de luz y la posición de la imagen a reconocer, por lo cual su eficiencia se ve comprometida al aplicarse sobre un sistema que se traslada como es el caso de un teléfono celular, por lo tanto si se lo desea aplicar en dichos dispositivos se lo debe hacer utilizando filtros y técnicas de Procesamiento Digital de Imágenes que aumenten la calidad de la imagen a reconocer.
- Al implementarse algoritmos que involucran Procesamiento Digital de Imágenes dentro de dispositivos móviles se sobrecarga al procesador con cálculos extensos por lo cual la respuesta de dichos algoritmos podría tardar varios segundos.

## 7.2 RECOMENDACIONES

- Según las comparativas obtenidas de la investigación obtenemos que el sistema operativo Android, es una excelente plataforma para desarrollar aplicaciones, por ser software libre sin necesidad de adquirir licencias de programación o instalación evitándonos los altos costos por licencias y permisos, además porque se lo puede empezar a programar con tan solo un pequeño conocimiento de java bien sea en Linux o Windows en cualquier versión.
- Al empezar a trabajar con nuevas tecnologías es importante seguir la documentación provista por el desarrollador de esta tecnología, pero también es conveniente leer foros web especializados ya que en los mismos se pueden encontrar respuestas a las inquietudes que puedan surgir en el proceso de aprendizaje.
- Al desarrollar en Android se debe tener en cuenta los tipos de recursos que tendrá la aplicación que se desea crear ya que dependiendo de los mismos variara el tiempo de compilación de la aplicación, dicho tiempo puede ser breve como durar varios minutos.
- Es recomendable que si se desea implementar algoritmos que impliquen gran capacidad computacional dentro de dispositivos móviles sean estos depurados de tal forma que se puedan aprovechar al máximo los recursos del dicho dispositivo.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] CONADIS, “Consejo Nacional de Discapacidades” <http://www.conadis.gob.ec>, consultado el 20 de Febrero de 2011.
- [2] Zhenwei, You, Touchphone For The Blind, <http://www.yankodesign.com/2009/06/15/touchphone-for-the-blind/>, 15 de Junio de 2009, consultado el 01 de Marzo de 2011.
- [3] Claudin, Rafa, Sens móvil táctil pero sin pantalla, [http://www.gizmodo.es/2008/06/18/sens\\_movil\\_tactil\\_pero\\_sin\\_pantalla.html](http://www.gizmodo.es/2008/06/18/sens_movil_tactil_pero_sin_pantalla.html), 18 de Junio de 2008, consultado el 01 de Marzo de 2011.
- [4] Polo, Juan, celular para ciegos en braille, <http://www.fayerwayer.com/2009/08/concepto-celular-para-ciegos-en-braille/>, 21 de Agosto de 2009, consultado el 01 de Marzo de 2011.
- [5] Mauro, Nokia Braille Reader hace posible que los ciegos lean SMS, <http://www.celularis.com/nokia/nokia-braille-reader-ciegos-sms.php>, 17 de Septiembre de 2009, consultado el 01 de Marzo de 2011.
- [6] Frost & Sullivan, Soluciones Corporativas para Smartphones, [http://mx.blackberry.com/services/business/smb/F&S%20White%20Paper%20RIM\\_20072010\\_ESP.pdf](http://mx.blackberry.com/services/business/smb/F&S%20White%20Paper%20RIM_20072010_ESP.pdf), 2010, consultado el 06 de Marzo de 2011.
- [7] Juanguis, Breve historia de los smartphones, <http://www.puntogeek.com/2011/01/14/breve-historia-de-los-smartphones/>, 14 de Enero de 2011, consultado el 06 de Marzo de 2011.
- [8] Guo, Chuanxiong, Wang, Helen y Zhu, Wenwu, “Smart-Phone Attacks and Defenses”, <http://research.microsoft.com/en-us/um/people/helenw/papers/smartphone.pdf>, consultado el 08 de Marzo de 2011.