



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TRABAJO DE GRADO, PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

TEMA:

**“MANUAL BÁSICO DE PRIMEROS AUXILIOS IMPLEMENTADO
EN LA PLATAFORMA ANDROID PARA TELÉFONOS
MÓVILES MEDIANTE ANIMACIONES CREADAS CON
EL SOFTWARE POSER”**

Autor: MARIO DANIEL MEDINA AGUILAR

Director: Ing. CARPIO AGAPITO PINEDA MANOSALVAS

Ibarra – Ecuador

2014



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DEL CONTACTO			
CÉDULA DE IDENTIDAD:	100313690-8		
APELLIDOS Y NOMBRES:	MEDINA AGUILAR MARIO DANIEL		
DIRECCIÓN:	ATUNTAQUI – SAN ROQUE		
EMAIL:	mdanielmedinaa@hotmail.com		
TELÉFONO FIJO:	06 2900 213 06 2900 004	TELÉFONO MÓVIL	0981191768
DATOS DE LA OBRA			
TÍTULO:	MANUAL BÁSICO DE PRIMEROS AUXILIOS IMPLEMENTADO EN LA PLATAFORMA ANDROID PARA TELÉFONOS MÓVILES MEDIANTE ANIMACIONES CREADAS CON EL SOFTWARE POSER		
AUTOR:	MARIO DANIEL MEDINA AGUILAR		
FECHA:	JUNIO 2014		
PROGRAMA:	PREGRADO		
TITULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES		
ASESOR / DIRECTOR:	ING. CARPIO AGAPITO PINEDA MANOSALVAS		

2.- AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Mario Daniel Medina Aguilar, con cédula de identidad Nro. 100313690-8, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

EL AUTOR:



MARIO DANIEL MEDINA AGUILAR

C.I.: 100313690-8



UNIVERSIDAD TÉCNICA DEL NORTE
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Mario Daniel Medina Aguilar, con cédula de identidad Nro. 100313690-8, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor de la obra o trabajo de grado denominado: Manual básico de primeros auxilios implementado en la plataforma Android para teléfonos móviles mediante animaciones creadas con el software Poser, que ha sido desarrollado para optar por el título de: Ingeniero en Sistemas Computacionales en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento en que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

A handwritten signature in blue ink, appearing to read 'Mario Daniel Medina Aguilar', is written over a horizontal line.

Nombre: Mario Daniel Medina Aguilar

Cédula: 100313690-8

En la ciudad de Ibarra Junio de 2014



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

CERTIFICACIÓN DEL ASESOR

Certifico que el trabajo de grado: Manual básico de primeros auxilios implementado en la plataforma Android para teléfonos móviles mediante animaciones creadas con el software Poser, ha sido realizado en su totalidad con interés profesional y responsabilidad por el señor: Mario Daniel Medina Aguilar, portador de la cédula de identidad Nro. 100313690-8.

Ing. Carpio Agapito Pineda Manosalvas

Asesor de Trabajo de Grado



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

En la ciudad de Ibarra, Junio de 2014

Nombre: Mario Daniel Medina Aguilar

Cédula: 100313690-8



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

DECLARACIÓN

Yo, Mario Daniel Medina Aguilar, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Técnica del Norte – Ibarra, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

En la ciudad de Ibarra, Junio de 2014

MARIO DANIEL MEDINA AGUILAR

C.I.: 100313690-8

CERTIFICACIÓN

Señores

Universidad Técnica del Norte

Presente

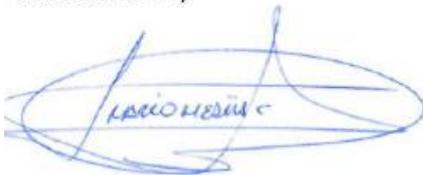
De mis consideraciones.-

Siendo auspiciantes del proyecto de tesis del señor Mario Daniel Medina Aguilar con cédula de identidad Nro. 100313690-8, quien desarrolló su trabajo con el tema: Manual básico de primeros auxilios implementado en la plataforma Android para teléfonos móviles mediante animaciones creadas con el software Poser, me es grato informar que se han superado con satisfacción las pruebas técnicas y la revisión de cumplimiento de los requisitos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte del señor Mario Daniel Medina Aguilar.

El señor Mario Daniel Medina Aguilar puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

En la ciudad de Ibarra, Junio 2014.

Atentamente,



Ing. Mario Medina Cruz

ANALISTA DE SISTEMAS

IESS – IMBABURA

CERTIFICACIÓN DE TRADUCCIÓN

Yo, Mariela Fernanda Castro Benavides, portadora de la cédula de identidad Nro. 100190575-9, en calidad de Docente del Área de Inglés de la Escuela de Educación Básica Jorge Peñaherrera, tengo a bien certificar que, el señor Mario Daniel Medina Aguilar, ha realizado la debida corrección con mi persona del Abstract de la tesis de grado con el tema: Manual básico de primeros auxilios implementado en la plataforma Android para teléfonos móviles mediante animaciones creadas con el software Poser, de la ciudad de Ibarra provincia de Imbabura el cual se encuentra bien estructurado, por lo que doy fe del presente trabajo.

Por tal motivo faculto a los peticionarios hacer uso del presente certificado como a bien consideren.



Lic. Mariela Fernanda Castro

C.I: 100190575-9



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

DEDICATORIA

Quiero ser agradecido y dedicar en especial a Dios, por darme la oportunidad de terminar una etapa más de mi vida, por darme las fuerzas necesarias para no desistir en el camino ante los problemas, a mis padres Mario y Pilar que me apoyaron siempre con sus buenos consejos, siempre estuvieron presentes con sus bendiciones, gracias por todos los recursos que me sirvieron para terminar mis estudios, a mis hermanas Cinthya y Tatiana, les quiero mucho gracias por sus ánimos en los momentos difíciles, gracias por compartir muchas cosas y por preocuparse de su hermano mayor, a mis sobrinos Mateo y Nicole que con su llegada han alegrado todo este tiempo a la familia pasando lindos momentos a su lado.

Daniel Medina



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

AGRADECIMIENTOS

A Dios, por guiarme por el buen camino, por darme la vida y sabiduría para alcanzar un logro más.

A mis padres, que han estado en todo momento apoyándome, guiándome para ser una persona de bien, humilde, con carácter, valores y principios.

A mi director de trabajo de grado Ingeniero Carpio Pineda, quien me brindó todo el tiempo con sus conocimientos y buenos consejos que guiaron a culminar con éxito el desarrollo de la presente tesis.

A la institución, por brindarme la oportunidad de estudiar y adquirir nuevos conocimientos que servirán en mi vida futura.

A los docentes, que compartieron con sus lecciones y experiencias para ser una persona de bien y preparada para cualquier adversidad en la vida profesional.

A todas las personas, que de una u otra forma estuvieron presentes para culminar este trabajo.

Daniel Medina

RESUMEN

En la actualidad existen muchas aplicaciones móviles que son de mucha utilidad para los usuarios que poseen ya sea un teléfono o una tablet, estas aplicaciones se las puede adquirir o descargar de forma gratuita o pagando un cierto costo, para poder disfrutar las funcionalidades que prestan.

Las aplicaciones móviles pueden ser de diferentes categorías tales como: juegos, redes sociales, música, noticias, entretenimiento, deportes, salud, vídeo/películas, mapas/navegación/búsqueda, entre otros.

Existen algunos sistemas operativos móviles como: Android, iOS, Windows Phone, Symbian y BlackBerry; cada uno de ellos tienen tiendas de aplicaciones que operan en línea, donde se puede buscar, descargar e instalar la aplicación que cumpla con las expectativas del usuario.

El presente proyecto busca conocer y comprender las características y el funcionamiento tanto del sistema operativo Android y el programa Poser Pro 2012. Además, comprende el desarrollo de una aplicación de nombre Primeros Auxilios que permite visualizar videos o animaciones de cómo se deben ejecutar éstas técnicas y procedimientos de carácter inmediato.

ABSTRACT

Nowadays there are a lot mobile applications which are very useful for users who already have a phone or a tablet. These applications can be acquired or downloaded free or by paying a true cost, so we can enjoy the features that they provide us.

The mobile applications can be of different categories such as: games, social networks, music, news, entertainment, sports, health, videos/films, maps/surf/search and so on.

There are some mobiles operating systems like: Android, iOS, Windows Phone, Symbian and BlackBerry; each one of them have shops of applications that operate in line, where we can search, download and install the application that satisfies with the user expectations.

This project seeks to know and understand the features and the performance, both Android operating system and the Poser Pro 2012 program. Besides it understands the development of an application with the name First aids that allow viewing videos or animation and how these techniques should be executed and procedures with character immediately.

ÍNDICE DE CONTENIDOS

AUTORIZACIÓN DE USO Y PUBLICACIÓN	ii
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO..	¡Error! Marcador no definido.
CERTIFICACIÓN DEL ASESOR	¡Error! Marcador no definido.
CONSTANCIAS	¡Error! Marcador no definido.
DECLARACIÓN	¡Error! Marcador no definido.
CERTIFICACIÓN	¡Error! Marcador no definido.
CERTIFICACIÓN DE TRADUCCIÓN	¡Error! Marcador no definido.
DEDICATORIA	x
AGRADECIMIENTOS	xi
RESUMEN	xii
ABSTRACT	xiii
ÍNDICE DE CONTENIDOS	xiv
INTRODUCCIÓN	xx
A. Problema	xx
B. Objetivos	xxi
C. Alcance y limitaciones	xxi
D. Justificación	xxii
CAPÍTULO I	1
DISPOSITIVOS MÓVILES	1
1.1 Definición	1
1.2 Clasificación	1
1.2.1 Dispositivos de comunicación	1
1.2.2 Dispositivos de computación	2
1.2.3 Reproductor Multimedia	2
1.2.4 Grabador multimedia	2
1.2.5 Consola portátil	2
1.3 Teléfonos móviles	2
1.4 Smartphone	3
1.5 Sistemas operativos móviles	4
1.5.1 Android	5
1.5.2 IOS	5
1.5.3 BlackBerry	5
1.5.4 Simbian	6

1.5.5 Windows Phone	6
CAPÍTULO II.....	7
ANDROID Y POSER	7
2.1 Android	7
2.1.1 Características	10
2.1.2 Arquitectura	10
2.1.3 Entorno de desarrollo.....	14
2.1.4 Estructura de un proyecto	21
2.1.5 Componentes de una aplicación	25
2.1.6 SQLite.....	33
2.1.7 Comunidad de desarrolladores Android	36
2.2 Software Poser	36
2.2.1 Introducción a Poser	36
2.2.2 Pose Room	37
2.2.3 Poser Library	46
CAPÍTULO III.....	49
DISEÑO Y DESARROLLO DE LA APLICACIÓN.....	49
3.1 Metodología de desarrollo ASD	49
3.1.1 Introducción	49
3.1.2 Características	49
3.1.3 Ciclo de vida	49
3.1.4 Ventajas y desventajas	50
3.2 Fases de la metodología	50
3.2.1 Fase Especular	50
3.2.1.1 Misión	50
3.2.1.2 Limitaciones	51
3.2.1.3 Ciclo 1 (Plan de ciclos).....	51
3.2.1.4 Ciclo 2 (Análisis)	52
3.2.1.5 Ciclo 3 (Diseño)	52
3.2.1.6 Ciclo 4 (Implementación)	53
3.2.1.7 Ciclo 5 (Pruebas y documentación).....	53
3.2.2 Fase Colaborar	54
3.2.2.1 Análisis	54
3.2.2.2 Diseño.....	61
3.2.2.3 Implementación.....	64
3.2.3 Fase Aprender	68

3.2.3.1 Pruebas	68
3.2.3.2 Documentación	71
CAPÍTULO IV	72
CONCLUSIONES Y RECOMENDACIONES	72
4.1 Análisis de costo beneficio	72
4.2 Impactos	74
4.2.1 Impacto social	74
4.2.2 Impacto ambiental.....	74
4.2.3 Impacto tecnológico	74
4.3 Conclusiones	75
4.4 Recomendaciones	75
BIBLIOGRAFÍA.....	77
PUBLICACIÓN EN LINEA	77
ANEXOS.....	84
ANEXO A MANUAL TÉCNICO	84
ANEXO B MANUAL DE USUARIO	96

ÍNDICE DE FIGURAS

Figura 1: Smartphone Sony Ericsson Xperia Play.....	3
Figura 2: Dominio de Android en el mercado mundial.....	5
Figura 3: Crecimiento de Android en activaciones de dispositivos	7
Figura 4: Fundadores, historia de Android	8
Figura 5: Versiones de Android.....	9
Figura 6: Arquitectura de Android	11
Figura 7: Descarga del JDK.....	14
Figura 8: Descarga de Eclipse	15
Figura 9: Configuración de la ruta del SDK de Android.....	15
Figura 10: Descarga del plugin ADT para Eclipse.....	16
Figura 11: Configuración de la ruta donde se encuentre el SDK.....	16
Figura 12: SDK de Android Manager	17
Figura 13: Creación y configuración del dispositivo virtual AVD.....	18
Figura 14: Creación de un proyecto Android.....	18
Figura 15: Configuración de datos de un proyecto Android.....	19
Figura 16: Creación y configuración del icono del proyecto	19
Figura 17: Creación de la actividad principal.....	20
Figura 18: Descarga del SDK y ADT Bundle para Windows	21
Figura 19: Carpeta /src	21
Figura 20: Carpeta /gen.....	22
Figura 21: Carpeta /bin	23
Figura 22: Carpeta /libs.....	23
Figura 23: Pila de Actividades de una aplicación Android	25
Figura 24: Ciclo de vida de una Activity	26
Figura 25: Árbol de jerarquía de View	29
Figura 26: Ciclo de vida del Servicio	31
Figura 27: Poser's Menu Bar	38
Figura 28: Light Controls.....	38
Figura 29: Camera Controls.....	39
Figura 30: Room Tabs	39
Figura 31: Editing Tools.....	41
Figura 32: Document Window.....	42
Figura 33: Display Controls.....	42
Figura 34: Parameters Palette	43

Figura 35: UI Dots.....	43
Figura 36: Animation Controls.....	45
Figura 37: Library Window	46
Figura 38: Ciclo de vida de la metodología ASD	49
Figura 39: Proceso visualizar animaciones	55
Figura 40: Proceso crear nuevo contacto.....	56
Figura 41: Proceso llamar contacto.....	58
Figura 42: Proceso editar contacto	59
Figura 43: Proceso eliminar contacto.....	60
Figura 44: Diagrama de datos módulo información del lesionado	62
Figura 45: Diagrama arquitectónico	62
Figura 46: Diagrama de navegación	63
Figura 47: Bocetos de las pantallas de la aplicación	64
Figura 48: Diagrama de clases	65
Figura 49: Interfaces valoración del lesionado	66
Figura 50: Interfaces de información del accidentado	66
Figura 51: Interfaces de reanimación cardio pulmonar.....	67
Figura 52: Interfaces de fracturas	67
Figura 53: Interfaces de quemaduras	67
Figura 54: Interfaces de botiquín de emergencia	68

ÍNDICE DE TABLAS

Tabla 1: Descripción del contenido de la carpeta /res	24
Tabla 2: Estados de una Activity	26
Tabla 3: Métodos que tiene una Activity.....	27
Tabla 4: Tipos de Intents	28
Tabla 5: Tipos de Layouts.....	29
Tabla 6: Ciclo de planificación	51
Tabla 7: Ciclo de análisis	52
Tabla 8: Ciclo de diseño	52
Tabla 9: Ciclo de implementación	53
Tabla 10: Ciclo de pruebas y documentación	53
Tabla 11: Requerimientos de los módulos	54
Tabla 12: Participantes o equipo de la aplicación móvil	55
Tabla 13: Descripción del proceso visualizar las animaciones	56
Tabla 14: Descripción del proceso crear nuevo contacto	57
Tabla 15: Descripción del proceso llamar contacto	59
Tabla 16: Descripción del proceso editar contacto.....	60
Tabla 17: Descripción del proceso eliminar contacto	61
Tabla 18: Dispositivos móviles de pruebas	69
Tabla 19: Plan de verificación de los requerimientos	70
Tabla 20: Pruebas funcionales.....	71
Tabla 21: Costos de hardware	72
Tabla 22: Costos de software	72
Tabla 23: Costos de suministros	73
Tabla 24: Costos varios	73
Tabla 25: Costo total.....	73

INTRODUCCIÓN

A. PROBLEMA

• Antecedentes

Muchas veces en la vida nos topamos con ciertos momentos en los cuales una persona que se encuentra con alguna lesión, fractura, intoxicación, golpes, caídas y otras; y que se encuentra en lugares apartados de la ciudad necesita que se le dé los primeros auxilios.

Las emergencias se suelen dar en momentos menos esperados, por ejemplo una madre que observa que su hijo menor tiene un accidente, una persona que está pasando y mira un accidente de tránsito, un jugador de fútbol que observa a su amigo sufrir de alguna lesión grave, todas estas personas deben tener un conocimiento básico de primeros auxilios para brindar ayuda.

Los primeros auxilios, son medidas terapéuticas urgentes que se realizan a personas que han sufrido accidentes o enfermedades imprevistas hasta disponer de un procedimiento por expertos. La intención es calmar el dolor y la angustia del accidentado o enfermo y evitar que su estado se empeore.

• Situación actual

En la actualidad la mayoría de las personas tienen un desconocimiento que podría decirse parcial o total sobre los primeros auxilios, desconocen qué hacer en esa situación, no saben cómo actuar, qué decisión tomar, como afrontar esa realidad que acontece y poder ayudar en algo o mejor aún salvar la vida de una persona.

• Definición del problema

Existen momentos en los cuales una persona sufre un accidente o una enfermedad inesperada y necesita urgentemente la aplicación de los primeros auxilios ya que los primeros minutos son cruciales para salvar la vida de dicha persona, sin embargo la gran mayoría de las personas no tienen acceso a esta información que permita guiarse ante lo que se debe hacer en esta situación hasta que lleguen las personas con mayor experiencia y puedan atender al paciente.

• Prospectiva

Si no se da la realización de este proyecto las personas seguirán teniendo ese desconocimiento de los primeros auxilios no podrán prestar su ayuda a la persona que lo necesita porque inclusive esta persona podría ser un familiar al cual se le puede salvar la vida mientras vienen las personas que dan la atención médica directa y adecuada.

B. OBJETIVOS

• Objetivo general

Desarrollar una aplicación para los diferentes usuarios que poseen un teléfono móvil basado en la plataforma Android dando a conocer toda la información básica mediante animaciones sobre los primeros auxilios de una manera rápida y eficaz que permita ayudar a personas accidentadas mientras llega la atención médica preparada.

• Objetivos específicos

- Identificar y buscar los diferentes accidentes más comunes que se pueden suscitar en una persona.
- Establecer la información necesaria y concisa que guiará en el momento que tenga que aplicar los primeros auxilios.
- Puntualizar todas las herramientas necesarias que ayudarán a la elaboración de la aplicación.
- Evaluar la aplicación rigurosamente hasta conseguir el resultado deseado.
- Elaborar la documentación necesaria que servirá como soporte para el buen uso de la aplicación móvil.
- Elaborar las animaciones para cada módulo del manual de primeros auxilios con la finalidad de que los usuarios visualicen lo que se debe hacer en caso de algún accidente.
- Insertar las animaciones en la aplicación móvil para su correcto funcionamiento y brindar la información al usuario.
- Emplear la metodología ágil ASD (Adaptive Software Development) para la elaboración de la aplicación con la finalidad de obtener un software de calidad.

C. ALCANCE Y LIMITACIONES

• Alcance

La mayoría de los módulos ofrecerá vídeos o animaciones para que el usuario tenga una visión más clara de lo que tiene que hacer para aplicar los primeros auxilios. Las animaciones serán creadas mediante un software denominado Poser Pro 2012 el cual se lo dará a conocer más adelante.

La aplicación estará a disposición de los usuarios en el Android Market especificando las políticas que sugieren para poder subir la aplicación y luego distribuirla. En el caso de no poder subir la aplicación estará dispuesta en un servidor que está situado en la institución del IESS para poder descargarla.

• Limitaciones

La aplicación será de carácter informativo debido a que es una nueva tecnología pero en el transcurso de la investigación se añadirán nuevas opciones que aportarán al desarrollo de este proyecto.

La aplicación será de tipo off-line es decir que no se necesita estar conectado a una red para ver su funcionamiento, los usuarios la podrán descargar gratuitamente y la instalarán en su dispositivo.

D. JUSTIFICACIÓN

En la actualidad el crecimiento del número de personas que utilizan dispositivos móviles hace evidente la relevancia de implementar aplicaciones que satisfagan las necesidades de los usuarios; aplicaciones que pueden ser tanto para el entretenimiento, de educación, aplicaciones ofimáticas, de organización, de accesibilidad, aplicaciones web, entre otras, que brindan información de una manera rápida y eficaz.

Las emergencias pueden suceder en cualquier momento, y pueden ser de cualquier tipo, como por ejemplo: una vía respiratoria bloqueada puede matar a alguien en 3 o 4 minutos, pero una ambulancia puede demorarse más de 10 minutos en llegar. Es así que un simple procedimiento como abrir las vías respiratorias de la víctima puede salvarle la vida mientras llega la unidad de emergencias. Según las estadísticas, es más probable que se tenga que administrar los primeros auxilios a alguien que se conoce que a extraños. Saber qué hacer en esos casos permitirá actuar rápido si un accidente sucede.

Este proyecto beneficiará a las siguientes personas:

- El principal beneficiado será la persona a quien se le dé los primeros auxilios ya que se tratará de mantenerlo con vida y estable o ponerlo fuera de peligro hasta llevarlo a un médico más capacitado.
- La persona que tiene la aplicación en su teléfono móvil ya que así podrá tener información donde podrá guiarse ante lo que debe hacer frente a una situación en la cual una persona ha sufrido un accidente y necesita de ayuda.
- Los paramédicos también serán beneficiados ya que cuando ellos lleguen al lugar donde está la víctima a ésta la encontrarán con los signos vitales estables y ellos podrán ejecutar su trabajo con más serenidad al ver que se han aplicado correctamente los primeros auxilios.

Se justifica este proyecto en vista de que las personas no saben cómo ejecutar los primeros auxilios debido a la falta de información a una persona que lo necesita y que se encuentra en lugares aislados de la ciudad, la finalidad de este proyecto es presentar información en el teléfono móvil sobre los procedimientos que se aplican a los accidentados hasta disponer de personas profesionales.

Las herramientas informáticas que se van a utilizar para el desarrollo de la aplicación son las siguientes:

- El SDK de Android (Software Development Kit) proporciona las herramientas y las API's necesarias para empezar a desarrollar todo tipo de aplicaciones en dispositivos que cuentan con la tecnología Android, por medio de un emulador se puede testear las aplicaciones desarrolladas para cualquier versión.

- El IDE Eclipse es, en el fondo, únicamente un almacén (workbench) se puede agregar herramientas para programar en cualquier lenguaje con la implementación de los plugins adecuados.

- El plugin ADT de Android para Eclipse. Android ofrece un plugin para el IDE Eclipse llamado ADT (Android Development Tools), que da posibilidad de construir aplicaciones para Android, éste extiende las capacidades de Eclipse para permitir crear proyectos Android rápidamente, agregar componentes basados en la API (Application Programming Interface) Framework de Android, depurar aplicaciones usando las herramientas del SDK, inclusive la opción de importar y exportar APKs para distribuir la aplicación.

- Para la creación de las animaciones se utilizará el software Poser es una herramienta para la creación y animación de figuras, modelos en 3D, exactamente de protagonistas con forma humana. Posee un eficaz motor gráfico que gestiona las articulaciones y músculos para mostrar naturalidad en los movimientos utilizando las capacidades de los ordenadores actuales.

- La metodología de desarrollo de software que se utilizará será ASD (Adaptive Software Development – Desarrollo de Software Adaptable), es una metodología ágil, un modelo de implementación para desarrollo de software. Propone la adaptación al cambio y no luchar contra él. Se basa en la adaptación continua a circunstancias cambiantes. Se justifica esta metodología ya que las metodologías ágiles constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores, debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

CAPÍTULO I

DISPOSITIVOS MÓVILES

1.1 DEFINICIÓN

Un dispositivo móvil es un aparato de pequeño tamaño fácil de transportar, con conexión permanente a una red, con memoria limitada, cuenta con una cierta cantidad de computación y almacenamiento de datos.

En la actualidad los dispositivos móviles forman un grupo sumamente diverso, se puede agregar cualquier elemento de hardware y software que amplía y varía su función inicial.

El más frecuente sin duda es la conexión telefónica, esto incluye el envío de SMS¹, MMS² y acceso a WAP³ o la conexión a Internet.

Igualmente están presentes la cámara fotográfica y video, receptor de radio, pantalla táctil, localizador GPS⁴, teclado QWERTY⁵, dispositivos de memoria extraíbles, acelerómetro y otros.

Desde el punto de vista del software, se pueden añadir un amplio número de aplicaciones como: navegadores web o clientes de correo, programas ofimáticos, videojuegos, reproductores de audio y video, entre otros.

1.2 CLASIFICACIÓN

Los dispositivos móviles se clasifican en:

1.2.1 DISPOSITIVOS DE COMUNICACIÓN



Un dispositivo de comunicación es aquel dispositivo cuya función principal es ofrecer una infraestructura de comunicación, principalmente telefónica. Estos dispositivos ofrecen además servicios como el envío de mensajes MMS y SMS, o acceso WAP.

¹ SMS. Short Message Service – Servicio de mensajes cortos.

² MMS. Multimedia Messaging System – Sistema de mensajería multimedia.

³ WAP. Wireless Application Protocol – Protocolo de aplicaciones inalámbricas.

⁴ GPS. Global Positioning System – Sistema de posicionamiento global.

⁵ QWERTY. Distribución de teclado más común, el nombre proviene de las primeras 6 letras de la fila superior.

1.2.2 DISPOSITIVOS DE COMPUTACIÓN



Los dispositivos de computación son aquellos dispositivos móviles que brindan mayores contenidos de procesamiento de datos, presentan una pantalla y teclado más cercanos a un ordenador de escritorio. Un dispositivo de computación también es un computador portátil o laptop, estos dispositivos presentan mayores beneficios de hardware igualando o superando a los ordenadores de escritorio. [WEB1]

1.2.3 REPRODUCTOR MULTIMEDIA



Dispositivo móvil diseñado específicamente para proporcionar al usuario la reproducción de contenidos audiovisuales; como audio, video o imágenes.

1.2.4 GRABADOR MULTIMEDIA



Es un dispositivo que permite la grabación de datos específicamente formato digital, ya sea de audio como las cámaras fotográficas digitales o video como las cámaras de video digital.

1.2.5 CONSOLA PORTÁTIL



Este dispositivo móvil tiene como función única la de proporcionar al usuario jugar videojuegos. Las consolas portátiles y los teléfonos fueron los primeros dispositivos móviles en convertirse en un producto de multitudes.

1.3 TELÉFONOS MÓVILES

Un teléfono móvil es un dispositivo inalámbrico electrónico que permite tener acceso a la red de telefonía celular o móvil. Su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado funciones adicionales como mensajería instantánea, agenda, juegos, cámara fotográfica y de vídeo, acceso a Internet, reproducción de video e incluso GPS y reproductor mp3.

Inicialmente los teléfonos móviles sólo permitían realizar llamadas de voz y enviar mensajes de texto. Conforme fue avanzando la tecnología se incluyeron nuevas aplicaciones como juegos, alarma, calculadora y acceso WAP (acceso a Internet mediante páginas web especialmente diseñadas para móviles).

La evolución del teléfono móvil ha permitido disminuir su tamaño y peso, por eso se ha llevado a cabo el desarrollo de baterías más pequeñas y de mayor duración, pantallas más nítidas y de colores, la incorporación de software más amigable.

1.4 SMARTPHONE

Un smartphone (teléfono inteligente), es una evolución del teléfono móvil normal que posee ciertas características y beneficios que lo aproximan más a un ordenador personal que un simple teléfono.

Entre las características, está el almacenamiento de datos y un incremento en la capacidad de proceso, la conexión a Internet mediante Wi-Fi, pantalla táctil, acelerómetro, posicionador geográfico, teclado QWERTY y aplicaciones como: navegador web, aplicaciones ofimáticas, reproductores de video y audio, entre otros, inclusive con la opción de descargar e instalar otras.



Figura 1: Smartphone Sony Ericsson Xperia Play

Fuente web: http://blog.xmundo.net/wp-content/uploads/2011/02/Xperia-PLAY_Black_screen11-300x290.png

A pesar de estas mejoras con respecto a sus antecesores móviles, el reducido tamaño de los smartphones conlleva inevitablemente limitaciones de hardware que los diferencian de ordenadores convencionales, limitaciones como: pantallas más pequeñas, menor contenido del procesador, prohibiciones de memoria RAM⁶ y memoria persistente, y la adaptación de una batería para el consumo de energía.

Las limitaciones anteriores son indispensables para verificar la capacidad existente del dispositivo y el desarrollo de software (aplicaciones de usuario o del propio sistema operativo).

1.5 SISTEMAS OPERATIVOS MÓVILES

Un sistema operativo móvil debe ser íntegro con una gran seguridad, los dispositivos con esas características no deben presentar reinicios o caídas. Además, debe adaptarse adecuadamente a las limitaciones de memoria y procesamiento de datos, facilitando la ejecución exacta y continuamente rápida al usuario.

Estos sistemas deben estar libres de errores antes de ingresar a producción. Un dispositivo móvil tiene las limitaciones de actualizar o reinstalar versiones mejoradas.

Un dispositivo móvil puede funcionar durante semanas o meses sin ser apagado y reiniciado, esto no pasa con un ordenador personal. El consumo de energía proporcionado por la batería debe ser productivo por ser limitada.

Todos estos aspectos de los dispositivos móviles, entre otros muchos, se toman en cuenta al momento de desarrollar un sistema operativo, tanto para los fabricantes como a los usuarios.

En la Figura 2 se observa el gran porcentaje que tiene Android en el mercado mundial de los sistemas operativos de teléfonos móviles entre abril y junio del 2012, se considera que lidera siendo el sistema operativo con más participación en el mercado. **[WEB2]**

- Android: 104.8 millones de unidades; 68,1% de participación en el mercado.
- iOS: 26 millones vendidos; 16,9% de participación.
- BlackBerry: 7.4 millones; 4,8%
- Symbian: 6.8 millones; 4,4%
- Windows: 5.4 millones; 3,5%

⁶ RAM. Random Access Memory – Memoria de Acceso Aleatorio.

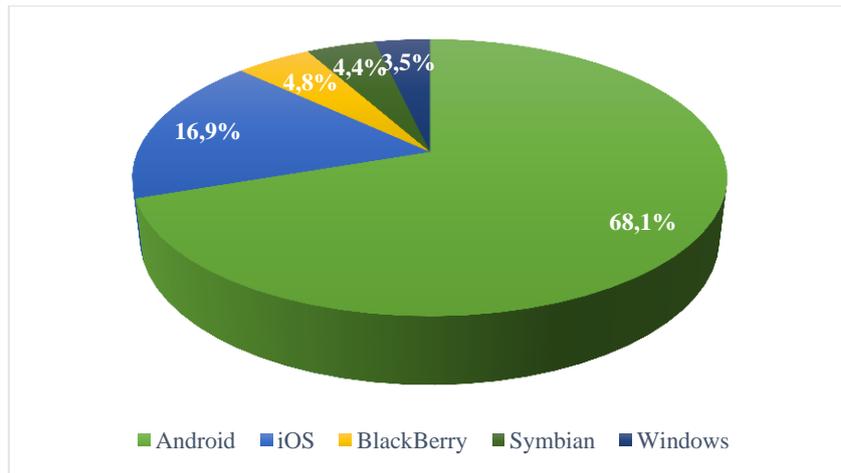


Figura 2: Dominio de Android en el mercado mundial

Fuente: propia

Hoy en día, existen algunos sistemas operativos para los dispositivos móviles. A continuación se detallan cinco de los sistemas operativos móviles:

1.5.1 ANDROID



Sin duda el líder del mercado móvil en sistemas operativos, está basado en Linux diseñado originalmente para dispositivos móviles como los teléfonos inteligentes pero después tuvo modificaciones para ser usados en tablets, hoy por hoy está desarrollo para usarse en netbooks y PC's, Google es el desarrollador de Android. **[WEB3]**

1.5.2 IOS



Creado por Apple básicamente para el iPhone a continuación usado en el iPad e iPod Touch. Es un originario de Mac OS X, se lanzó en el año 2007. Su principal revolución es una combinación casi perfecta el hardware y software. **[WEB3]**

1.5.3 BLACKBERRY



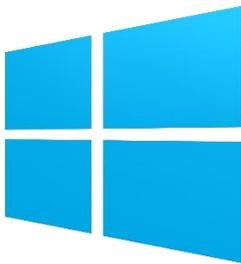
Este sistema operativo fue desarrollado por la compañía RIM (Research In Motion), cuya característica principal se centra en ser un software diseñado fundamentalmente para uso profesional, gracias a las herramientas correo electrónico y agenda. Brinda una gran funcionalidad en el ámbito profesional, también da la posibilidad de tener acceso a las cuentas de correo electrónico y navegador por Internet en tiempo real.

1.5.4 SIMBIAN



Es un sistema operativo móvil abierto y estándar para dispositivos de telefonía móvil. Este sistema se formó de la alianza entre varias empresas, siendo Nokia la más importante, Sony Ericsson, Samsung, Motorola en un momento ésta alianza permitió ser unos de los fundadores y más usados, pero actualmente está perdiendo paulatinamente la cuota de usuarios.

1.5.5 WINDOWS PHONE



Es un sistema operativo móvil compacto desarrollado por Microsoft, cuenta con un conjunto de aplicaciones básicas, actualmente va por la versión 7. Diseñado para poseer una similitud a las versiones de escritorio de Windows estéticamente, llamado anteriormente como Windows Mobile. **[WEB3]**

CAPÍTULO II

ANDROID Y POSER

2.1 ANDROID

Android domina en cientos de millones de dispositivos móviles en más de 190 países de todo el mundo. Es la mayor plataforma móvil instalada, su rápido crecimiento de millones de usuarios con dispositivos Android empiezan por primera vez a buscar aplicaciones, juegos, y otros contenidos digitales. **[WEB4]**

Android ofrece una plataforma de clase mundial para la creación de aplicaciones y juegos a todos los usuarios de Android, así como un mercado abierto para la distribución de éstos de forma instantánea. **[WEB4]**



Figura 3: Crecimiento de Android en activaciones de dispositivos

Fuente web: <http://developer.android.com/images/about/growth-chart.png>

En la Figura 4 se da a conocer en forma resumida la historia de Android desde su creación en el año 2003, donde la empresa Google en el 2007 anuncia oficialmente el desarrollo de Android OS.

En la Figura 5 se muestra la evolución de todas las versiones, comenzando por la 1.0 OS llegando hasta Android 4.1 Jelly Bean y una versión en la actualidad que está a punto de lanzarse Android 4.4 Kit Kat. **[WEB5]**

Octubre de 2003
ANDROID INC. fundada

por Andy Rubin, Rich Miner, Nick Sears y Chris White



Android era el apodo de Andy Rubin debido a su fascinación por los robots

“Pensamos que sería bueno si podríamos construir una plataforma de cámara con aplicaciones de terceros,”
- Andy Rubin

ORIGINALMETE DISEÑADO PARA CÁMARAS DIGITALES

Enfocó a smartphones después de decidir que las cámaras no eran un gran mercado suficiente.

Rubin agotó el dinero en 2003, pero Steve Perlman Silicon Valley inventor y empresario, le dio en efectivo \$10,000 para que continúe con su trabajo.



2005

ANDROID INC. COMPRADA POR GOOGLE

Parte del esfuerzo de Google para ganar más tráfico móvil “adquirimos Android debido a los talentosos ingenieros y gran tecnología. Estamos encantados de tenerlos aquí.

- Portavoz de Google”



05 de Noviembre de 2007

GOOGLE ANUNCIA OFICIALMENTE EL DESARROLLO DEL SISTEMA OPERATIVO ANDROID

llega a ser conocido como el cumpleaños oficial de Android. Las operaciones Android Inc, seguía siendo un secreto, hasta este punto Android sería 100% libre y de código abierto, en contraposición estricta y regulado por iOS de Apple.



Figura 4: Fundadores, historia de Android

Fuente web: <http://www.startapp.com/infographics/history-of-android.aspx>



Figura 5: Versiones de Android

Fuente web: <http://www.startapp.com/infographics/history-of-android.aspx>

Según (Báez, y otros, Introducción a Android, 2012, pág. 1) manifiesta:

“Android, es un sistema operativo y una plataforma software, basado en Linux para teléfonos móviles. Además, también usan este sistema operativo (aunque no es muy habitual), tablets, netbooks, reproductores de música e incluso PC’s.”

Con Android se pretende que el desarrollador pueda utilizar al máximo las diferentes funcionalidades que brinda un dispositivo móvil (llamadas, mensajes de texto, cámara, agenda de contactos, conexión Wi-Fi, Bluetooth, aplicaciones ofimáticas, videojuegos, entre otros), para crear aplicaciones que satisfagan las necesidades de los usuarios. Dicho de otra manera, se intenta que el desarrollo de aplicaciones sea para cualquier dispositivo móvil. **[WEB6, pág. 37]**

2.1.1 CARACTERÍSTICAS

El sistema operativo móvil Android presenta las siguientes características: **[WEB7]**

- SQLite para almacenamiento de datos estructurados
- Soporte de formatos multimedia para audio, vídeo e imágenes (MPEG-4, H.264, MP3, AAC, OGG, AMR, JPEG, PNG, GIF).
- Conectividad (GSM, EDGE, 3G, Bluetooth y Wi-Fi).
- Soporte para hardware adicional (cámaras de vídeo, pantallas táctiles, GPS, acelerómetros).
- Diversidad de diseños (VGA, librerías de gráficos 2D y 3D basada en OpenGL y otros.).
- Máquina virtual de Java. Las aplicaciones escritas en lenguaje Java pueden ser compiladas y ejecutadas en la máquina virtual Dalvik, la cual es una máquina virtual optimizada para emplearla en dispositivos móviles.
- Entorno de desarrollo (emulador, herramientas de depuración, perfiles de memoria y funcionamiento, plugin para el IDE Eclipse).
- SDK⁷ disponible de forma gratuita.

2.1.2 ARQUITECTURA

Android es un sistema operativo organizado en capas y bibliotecas con diferentes funcionalidades, servicios e interacciones con las demás partes o subsistemas del sistema operativo.

⁷ SDK. Software Development Kit. Kit de Desarrollo Android

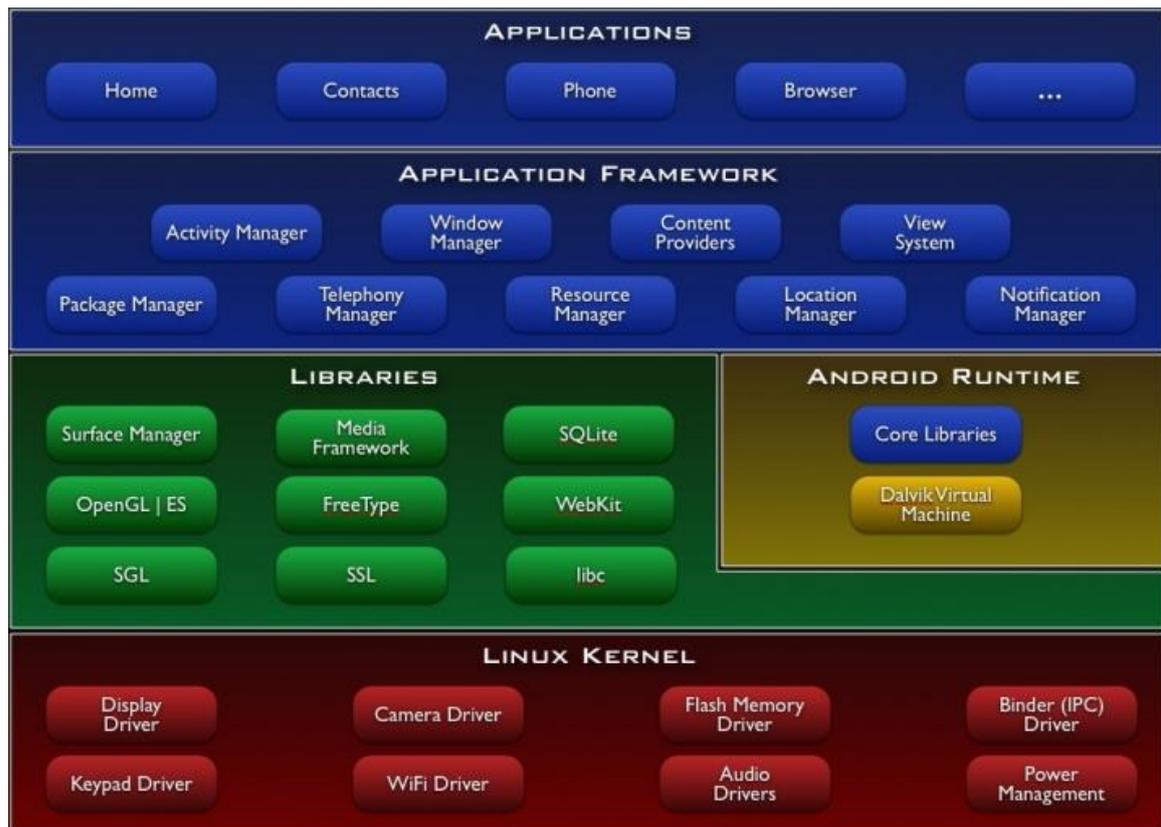


Figura 6: Arquitectura de Android

Fuente web: http://elinux.org/Android_Architecture

• Linux Kernel

El núcleo es el encargado de gestionar todos los recursos del sistema (hardware y sus funciones). Al desarrollador no se le permite el acceso directo a esta capa, debe hacer el uso de las librerías disponibles en las capas superiores. De esta forma también se evita el hecho de romperse la cabeza para conocer las características precisas de cada teléfono. Por ejemplo si se desea hacer uso de la cámara, el encargado del uso es el sistema operativo. Cualquier elemento de hardware existente en el teléfono, el kernel posee un controlador para utilizarlo desde el software.

• Libraries

Contiene un conjunto de librerías escritas en C o C++, antes de lanzar a la venta el dispositivo, se instalan las librerías creadas por el fabricante; el propósito es prestar funcionalidad en las tareas que son ejecutadas con continuidad por parte de las aplicaciones, con esto se evita estar codificando una y otra vez. De esta forma las aplicaciones son más eficientes.

• **Android Runtime**

La máquina virtual Dalvik, es el principal componente del entorno de ejecución de Android, donde se ejecutan cada una de las aplicaciones no originarias de Android. Son codificadas con el lenguaje de programación Java y compiladas en un formato concreto para que sean ejecutadas por Dalvik. Presentan una ventaja, la de compilar una sola vez, con esto las aplicaciones pueden comercializarse asegurando la ejecución en cualquier dispositivo Android que tenga como versión mínima del sistema operativo que requiera la aplicación. **[WEB8]**

Cabe comentar que, los archivos ejecutables generados con el SDK de Android tienen la extensión `.dex` que es específico para Dalvik, ésta es la razón por la que no se pueden ejecutar aplicaciones Java en Android ni viceversa. **[WEB8]**

• **Application Framework**

Capa que presenta las clases y servicios que manejan solamente las aplicaciones para efectuar sus funciones. La mayor parte de componentes son librerías Java, permitiendo acceder a recursos de capas preliminares a través de Dalvik. Siguiendo la Figura 6 se tiene lo siguiente: **[WEB8]**

- **Activity Manager:** gestiona la pila de actividades de la aplicación y ciclo de vida.
- **Windows Manager:** organiza lo que tiene que mostrar en pantalla. Establece las áreas en la pantalla que más adelante ocuparán las actividades.
- **Content Provider:** permite que los datos sean compartidos entre aplicaciones para tener control sobre cómo se accede a la información.
- **Views:** permite la elaboración de interfaces de usuario como: listas, botones, cajas de texto, entre otros; así como elementos aún más avanzados ya sea un visor de Google Maps o un navegador.
- **Notification Manager:** servicios de notificaciones que indican al usuario para solicitar de su atención manifestando alertas en la barra de estado. Permite añadir sonidos, activar la función de vibración o usar las luces LED⁸ del teléfono si los tiene.
- **Package Manager:** permite obtener información de aplicaciones instaladas en el dispositivo, además administrar la instalación de nuevas. Las aplicaciones contienen el

⁸ LED. Light Emmitting Diode – Diodo Emisor de Luz

archivo .apk, éstos a su vez contienen los archivos .dex con todos los recursos que requiera la aplicación, para proveer su descarga e instalación.

- **Telephony Manager:** ayuda a realizar llamadas o enviar y recibir mensajes de texto o mensajes multimedia, pero no permite eliminar la actividad que se abre cuando una llamada está en curso.

- **Resource Manager:** administra todos los elementos que integran parte de la aplicación y no están dentro del código, como: sonidos, cadenas de texto traducidas a diferentes idiomas, imágenes o layouts.

- **Location Manager:** permite determinar la posición geográfica del dispositivo Android mediante GPS o redes disponibles y trabajar con mapas.

- **Sensor Manager:** permite manejar los elementos de hardware del teléfono como acelerómetro, sensor de luminosidad, brújula, sensor de presión, sensor de proximidad, sensor de temperatura entre otros.

- **Camera:** permite el uso de la cámara o cámaras del dispositivo para tomar fotografías o grabar video.

- **Multimedia:** permiten reproducir y visualizar audio, vídeo e imágenes en el dispositivo.

- **Applications**

En esta última capa se encuentran las aplicaciones del dispositivo móvil, como: las que poseen interfaz de usuario o no, las dispuestas en C o C++ y las programadas en Java, las que vienen preinstaladas por defecto y las que ya han sido instaladas por parte del usuario.

También está presente la aplicación principal del sistema: Inicio (Home) o lanzador (launcher), permite ejecutar otras aplicaciones mediante una lista, mostrando diferentes escritorios, con la posibilidad de crear accesos directos a aplicaciones o incluso widgets, que también son aplicaciones que forman parte de esta capa.

Cabe mencionar que todas las aplicaciones, sean nativas de Android, aplicaciones de Google, las el fabricante incluye de serie del teléfono o las que son instaladas por el usuario manejan el similar marco de aplicación para acceder a los servicios que facilita el sistema operativo. **[WEB8]**

2.1.3 ENTORNO DE DESARROLLO

Se describen los requisitos necesarios para disponer de las herramientas y del entorno para desarrollar aplicaciones Android. Existen muy buenos tutoriales sobre la instalación de Eclipse y Android, incluida la documentación oficial de la plataforma.

La configuración se lo hará en el sistema operativo Windows 7 (32 bits y 64 bits). Los requerimientos del sistema son los siguientes: **[WEB4]**

- **Sistema Operativo:** Windows XP (32 bits), Vista (32 o 64 bits), o Windows 7 (32 o 64 bits)

- **IDE Eclipse:** Eclipse 3.6.2 (Helios) o superior.

Nota: Eclipse 3.5 (Galileo) ya no es compatible con la última versión del ADT. Complemento Eclipse JDT “Java Development Tools” (incluido en la mayoría de los paquetes del IDE Eclipse).

JDK 6 (sólo JRE “Java Runtime Environment” no es suficiente).

Plugin ADT⁹ (recomendado)

• Descarga e instalación del JDK

El JDK ¹⁰ se descarga de la dirección:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

Se escoge JDK de acuerdo al sistema operativo, se marca la opción Accept License Agreement, con esto ya se podrá descargar para las versiones de Windows x86 y Windows x64. Para su instalación simplemente se dará clic en Next en todas las ventanas que se aparezcan y dar clic en Finish para finalizar la instalación.

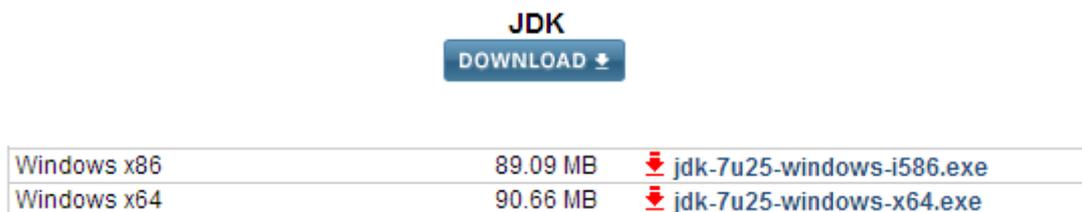


Figura 7: Descarga del JDK

Fuente: propia

⁹ ADT. Android Development Tools – Herramientas de Desarrollo Android

¹⁰ JDK. Java Development Kit – Kit de Desarrollo Java

• Descarga e instalación de Eclipse

Para descargar el IDE Eclipse se lo puede hacer de: <http://www.eclipse.org/downloads/>.

Se tiene que escoger la opción Eclipse IDE for Java EE Developers para Windows de 32 bits o Windows de 64 bits.

La instalación se debe descomprimir el ZIP en la ubicación deseada.



Figura 8: Descarga de Eclipse

Fuente: propia

• Descarga del SDK

Se lo puede descargar de: <http://developer.android.com/intl/es/sdk/index.html>.

Se descomprime el ZIP en cualquier ubicación, pero en este caso se descomprimirá el archivo en la siguiente ubicación → C:\android-sdk.

Para la configuración de la ruta del SDK Manager de Android en la variable de entorno PATH se hace lo siguiente: Equipo → Propiedades (botón derecho del ratón) → Configuración avanzada del sistema → Variables de entorno → Editar la variable PATH añadiendo al final punto y coma [;] en caso de no estar puesto la ruta del SDK Manager de Android.

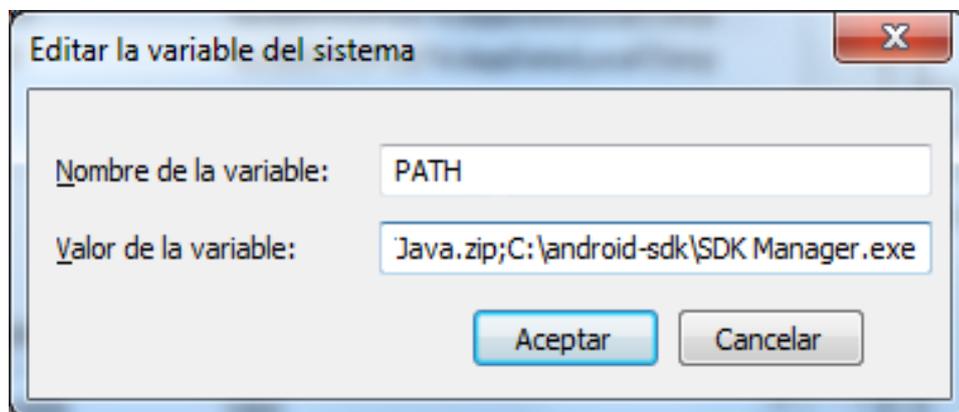


Figura 9: Configuración de la ruta del SDK de Android

Fuente: propia

• Descarga del plugin Android

Según (Lee, 2011, pág. 38) manifiesta:

“El ADT es una extensión para Eclipse IDE que soporta la creación y depuración de aplicaciones Android.”

El plugin para Eclipse que ofrece Google a los desarrolladores se llama Android Development Tools (ADT), brinda una gran ayuda en el desarrollo de aplicaciones.

Para descargar se tiene que ir a las opciones de actualización de Eclipse, accediendo al menú Help → Install New Software, ingresando: <https://dl-ssl.google.com/android/eclipse/>.

Nota: Si se presenta un problema para conectar el entorno con la URL indicada, se debe probar con http: eliminando la letra “s”.

Se selecciona el paquete completo formado por Android DDMS¹¹ y Android Developer Tools. **[WEB9, pág. 8]**

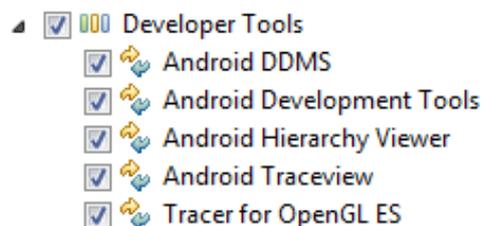


Figura 10: Descarga del plugin ADT para Eclipse

Fuente: propia

• Configuración del plugin ADT

En la ventana de configuración de Eclipse, se accede a la unidad de Android indicando la ruta donde se ha instalado el SDK. Esto se lo puede hacer accediendo al menú Window → Preferences.

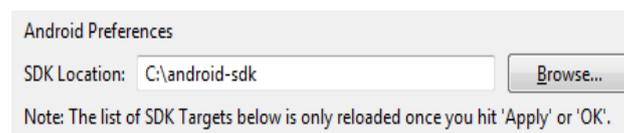


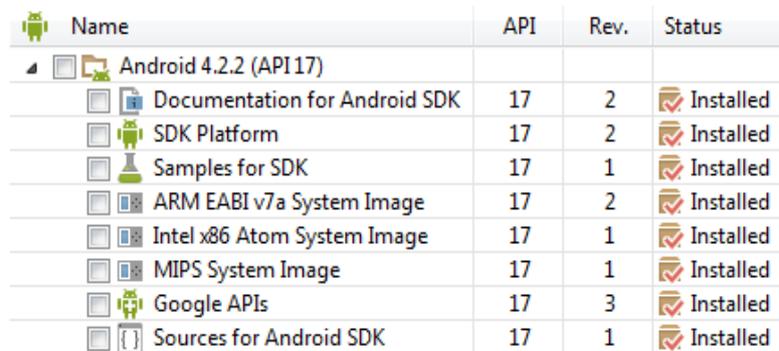
Figura 11: Configuración de la ruta donde se encuentre el SDK

Fuente: propia

¹¹ DDMS. Dalvik Debug Monitor Server – Servidor Monitor de Depuración Dalvik

• Descarga de los targets necesarios

Los targets son librerías necesarias para el desarrollo de aplicaciones en las diferentes versiones que presenta Android. Por ejemplo si se quiere desarrollar para Android 1.6 se tendrá que descargar el target correspondiente. Para ello desde Eclipse se accede al menú Window → Android SDK Manager. [WEB9, pág. 9]



Name	API	Rev.	Status
Android 4.2.2 (API 17)			
Documentation for Android SDK	17	2	Installed
SDK Platform	17	2	Installed
Samples for SDK	17	1	Installed
ARM EABI v7a System Image	17	2	Installed
Intel x86 Atom System Image	17	1	Installed
MIPS System Image	17	1	Installed
Google APIs	17	3	Installed
Sources for Android SDK	17	1	Installed

Figura 12: SDK de Android Manager

Fuente: propia

• Configuración de un AVD

Según (Lee, 2011, pág. 40) declara:

“Un AVD es una instancia de emulador que le permite modelar un dispositivo. Todo AVD consta de un perfil de hardware, un mapeo a una imagen del sistema, al igual que almacenamiento emulado, como una tarjeta de memoria SD¹².”

Conforme avanza el desarrollo de una aplicación no es necesario probar en un dispositivo real su funcionamiento, sino que se puede configurar un emulador o dispositivo virtual donde se puede realizar fácilmente estas tareas. Para ello se accede al menú Window → Android Virtual Device Manager, y con solo dar clic en New se logrará agregar los AVD que sean necesarios con la respectiva configuración que puede ser la versión, el tamaño de pantalla y otros. Para realizar la configuración de un AVD hay que indicar un nombre, el target de Android, y las características de hardware del dispositivo virtual, como: la resolución de pantalla, el tamaño de la tarjeta SD, GPS. [WEB9, pág. 10]

Para el tamaño de Internal Storage y SD Card se tiene las opciones KiB (Kibibyte), MiB (Mebibyte) y GiB (Gibibyte), son unidades de información utilizadas como múltiplos del byte, la equivalencia de estas unidades es la siguiente: [WEB10]

Byte → 1024 bytes = 1KiB Kibibyte → 1024 KiB = 1MiB Gibibyte → 1024 MiB = 1GiB

¹² SD. Secure Digital. Formato de tarjeta de memoria para dispositivos portátiles.

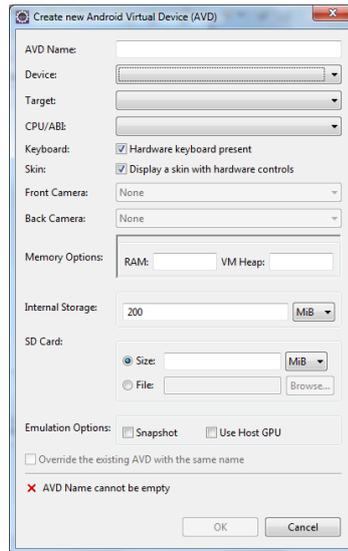


Figura 13: Creación y configuración del dispositivo virtual AVD

Fuente: propia

• Prueba de concepto “Hola Mundo”

Se crea un nuevo proyecto Android Project, indicando el nombre de la aplicación, del proyecto y del paquete, la versión mínima de Android, el target SDK, la versión de compilación y el tema. [WEB9, pág. 10]

Se tendrá que seleccionar además la versión mínima del SDK que aceptará la aplicación al ser instalada en un dispositivo, la versión del SDK para la que se desarrolla y la versión del SDK con la que se compilará el proyecto. Las dos últimas suelen coincidir con la versión de Android más reciente. El resto de las opciones se las dejará con los valores por defecto.

Para eso en el menú de Eclipse se debe ir a: File→New→Other→Android→Android Application Project.

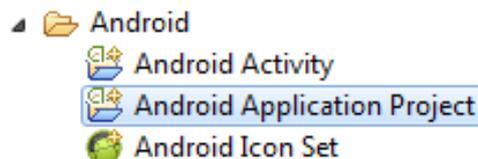


Figura 14: Creación de un proyecto Android

Fuente: propia

Application Name:

Project Name:

Package Name:

Minimum Required SDK:

Target SDK:

Compile With:

Theme:

Figura 15: Configuración de datos de un proyecto Android

Fuente: propia

Al pulsar Next, se accede al siguiente paso, donde se tendrá que indicar si durante la creación del nuevo proyecto se quiere crear un icono para la aplicación (Create Custom launcher icon) y si se quiere crear una actividad inicial (Create activity). También se podrá indicar si el proyecto será del tipo librería (Mark this Project as a Library). Pero para esta prueba de concepto se deja las opciones marcadas por defecto.

En la siguiente pantalla se puede configurar el icono que tendrá la aplicación en el dispositivo. Se puede seleccionar entre una imagen, un texto o un dibujo, al icono se lo puede editar cambiando el color de fondo, el margen, la forma, entre otras cosas más.

Foreground:

Image File:

Trim Surrounding Blank Space

Additional Padding:

Foreground Scaling:

Shape:

Background Color:

Figura 16: Creación y configuración del icono del proyecto

Fuente: propia

Luego de esto se debe elegir el tipo de actividad principal de la aplicación. Por ahora se entiende como actividad a una ventana o pantalla de la aplicación. Se deja todos los valores por defecto, indicando así que la pantalla principal será del tipo BlankActivity.

Se tiene que dar el nombre de la actividad principal, nombre del layout (interfaz gráfica de la actividad) y permite escoger las opciones de navegación como deslizar horizontal para la barra de acción.



The image shows a screenshot of the Android Studio interface during the creation of a new activity. It features three input fields: 'Activity Name' with the value 'Main', 'Layout Name' with the value 'main', and 'Navigation Type' which is a dropdown menu currently set to 'None'.

Figura 17: Creación de la actividad principal

Fuente: propia

• Elección de la plataforma de desarrollo

Al comenzar a desarrollar una aplicación, se debe analizar si se necesita alguna característica específica útil para una versión. Los usuarios con versiones anteriores no podrán instalar la aplicación en los dispositivos, por eso es recomendable seleccionar la menor versión posible que la aplicación pueda soportar.

Por ejemplo, si la aplicación necesita utilizar varios cursores simultáneos en la pantalla táctil (multi-touch), se tiene que elegir la versión 1.6 es la primera que lo soporta. Pero como se mencionó la aplicación no podrá ser instalada en versiones anteriores.

• ADT Bundle para Windows

Si con los pasos anteriores existen problemas al configurar el entorno de desarrollo, se puede descargar el paquete de ADT. Posee todas las herramientas necesarias para empezar y agilizar el desarrollo de aplicaciones Android. **[WEB4]**

El paquete ADT posee los siguientes componentes:

- Eclipse + ADT Plugin.
- Herramientas Android SDK.
- Plataforma de Herramientas Android.
- La última plataforma Android.
- La imagen actual del emulador de Android.



Figura 18: Descarga del SDK y ADT Bundle para Windows

Fuente web: <http://developer.android.com/images/sdk-cube.png>

2.1.4 ESTRUCTURA DE UN PROYECTO

Antes de empezar a desarrollar una aplicación es necesario conocer la estructura que la compone.

Al crear un nuevo proyecto se crea automáticamente la estructura de carpetas necesarias para luego ejecutar la aplicación. Ésta estructura será común a cualquier aplicación, independiente de su tamaño y complejidad.

A continuación se describe los elementos principales:

- **Carpeta /src**

Se encuentra presente el código fuente de: la aplicación, la interfaz gráfica, clases auxiliares, entre otros. Al inicio, Eclipse crea el código básico de la actividad principal de la aplicación, siempre bajo el paquete creado en el proyecto.

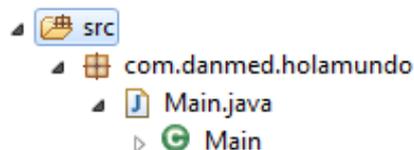


Figura 19: Carpeta /src

Fuente: propia

• Carpeta /gen

Contiene el código que se genera de forma automática, cada vez que el proyecto sea compilado. Al momento de generar el proyecto, la compilación de Android crea una serie de archivos fuente en java que hacen referencia al control de los recursos de la aplicación.

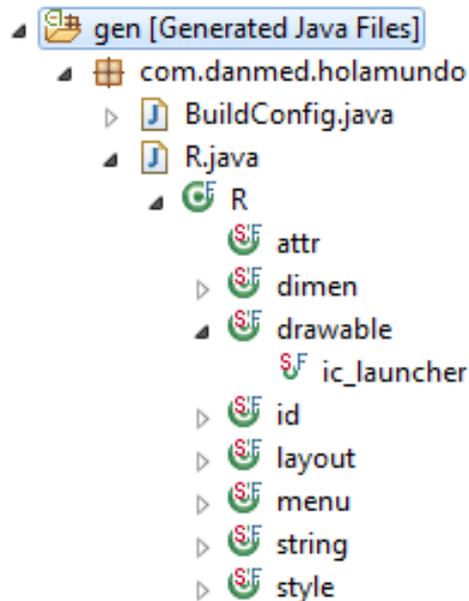


Figura 20: Carpeta /gen

Fuente: propia

El archivo R.java es el más importante. La clase R contiene un conjunto de constantes con los ID de los recursos que añaden en el directorio /res/, de forma que se pueda acceder fácilmente a estos recursos desde el código a través de este dato. Así, por ejemplo R.drawable.ic_launcher contendrá el ID de la imagen ic_launcher.png contenida en el directorio /res/drawable.

• Carpeta /assets

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

Dentro de la clase R.java se genera un ID para los recursos del directorio /res/raw/ se y se accede con los diferentes métodos de acceso de recursos y para los recursos del directorio /assets/ no se genera ID y se accede por medio de la ruta como cualquier otro fichero del sistema. Se usa uno u otro según las necesidades de la aplicación.

- **Carpeta /bin**

Los archivos que contiene esta carpeta no deben ser modificados. Contiene los elementos compilados de la aplicación y archivos complementarios, aquí se encuentra el archivo con extensión .apk que será instalado en el dispositivo.

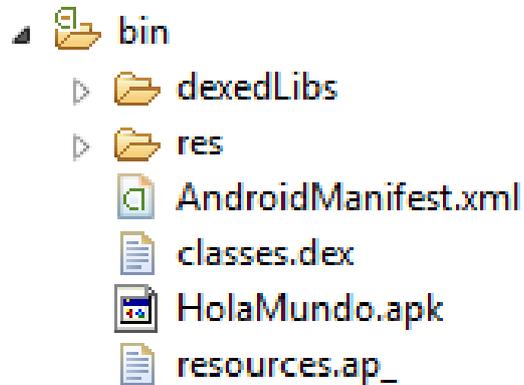


Figura 21: Carpeta /bin

Fuente: propia

- **Carpeta /libs**

Contiene las librerías auxiliares, normalmente en formato “.jar” que se vaya a utilizar en la aplicación.

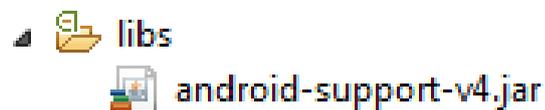


Figura 22: Carpeta /libs

Fuente: propia

- **Carpeta /res**

Contiene todos recursos de la aplicación como: imágenes, vídeos, cadenas de texto, entre otros. En la tabla 1 se detallan las subcarpetas donde se pueden añadir los recursos.

Subcarpeta	Descripción
/res/drawable/	<p>Contiene las imágenes y otros elementos gráficos usados en la aplicación. Los recursos pueden ser definidos para todo tipo de pantallas que poseen diferentes densidades:</p> <p>/drawable-ldpi (pantallas para densidad baja)</p> <p>/drawable-mdpi (pantallas para densidad media)</p> <p>/drawable-hdpi (pantallas para densidad alta)</p> <p>/drawable-xhdpi (pantallas para densidad muy alta)</p>
/res/layout/	<p>Contiene archivos XML de pantallas de la interfaz gráfica. Para crear layouts que dependen de la orientación del dispositivo se tiene que crear dos subcarpetas:</p> <p>/layout (vertical)</p> <p>/layout-land (horizontal)</p>
/res/anim/ /res/animator/	Contienen las animaciones que van a ser utilizadas por la aplicación.
/res/color/	Contiene ficheros XML de definición de colores.
/res/menu/	Contiene la definición XML de los menús de la aplicación.
/res/xml/	Contiene archivos XML de datos.
/res/raw/	Contiene recursos añadidos, que son diferentes al formato XML y que no están en las otras carpetas de recursos.
/res/values/	Contiene archivos XML como: strings.xml (cadenas de texto), styles.xml (estilos), colors.xml (colores), arrays.xml (arrays de valores).

Tabla 1: Descripción del contenido de la carpeta /res

Fuente web: <http://www.sgoliver.net/blog/?p=1278>

No todas éstas carpetas tienen por qué aparecer en cada proyecto Android, tan sólo las que se necesiten. Esto depende para que versión se está realizando el proyecto por ejemplo si se tiene values-v11 y values-v14, serán para todos los dispositivos que cuenten con la versión Android 3.0 o superior.

• **AndroidManifest.xml**

Es un archivo XML que contiene todos los elementos principales de la aplicación, entre los cuales están: la identificación (nombre, versión, icono y otros.), los componentes (pantallas, mensajes y otros.), las librerías complementarias usadas, o los permisos declarados para la ejecución.

2.1.5 COMPONENTES DE UNA APLICACIÓN

Para la creación de una aplicación Android, se debe conocer los elementos que posee y la funcionalidad de cada uno. El funcionamiento de Android se basa en Linux, donde cada aplicación usa su propio proceso. Tienen un identificador que los distingue, así al momento ejecutarse accedan solo a los archivos que le corresponden.

La ejecución principal en los dispositivos, es la aplicación visible en la pantalla, además puede tener algunas aplicaciones en segundo plano con su propia pila de tareas.

“La pila de tareas es la secuencia de ejecución de procesos en Android” (Báez, y otros, Introducción a Android, 2012, pág. 15)

Está compuesta de actividades apiladas según sean llamadas, al ser una pila de tareas las que se encuentran en la parte superior se finalizarán primero, si se necesita que las tareas de abajo finalicen el sistema se encarga de destruir porque se necesita memoria, tienen que estar dispuestas para cesar en cualquier instante.



Figura 23: Pila de Actividades de una aplicación Android

Fuente: propia

• **Activities (Actividades)**

Una actividad es la componente principal encargada de mostrar al usuario la interfaz gráfica, es decir, una actividad sería el equivalente a una ventana, y es el medio de comunicación entre la aplicación y el usuario. Se define una actividad por cada interfaz del proyecto.

- Representa una cosa concreta que puede hacer el usuario.
- Corresponden con una pantalla de la interfaz de usuario.
- Muestra los controles de la interfaz de usuario y reacciona ante las interacciones del mismo.
- Es una clase derivada de la clase Activity.
- Toda actividad se declara en el archivo AndroidManifest.xml.

- Normalmente una aplicación consta de varias actividades.
- Cada pantalla se implementa como una actividad.
- Moverse a la siguiente actividad supone llamar al método `startActivity()`, `startActivityForResult()`.

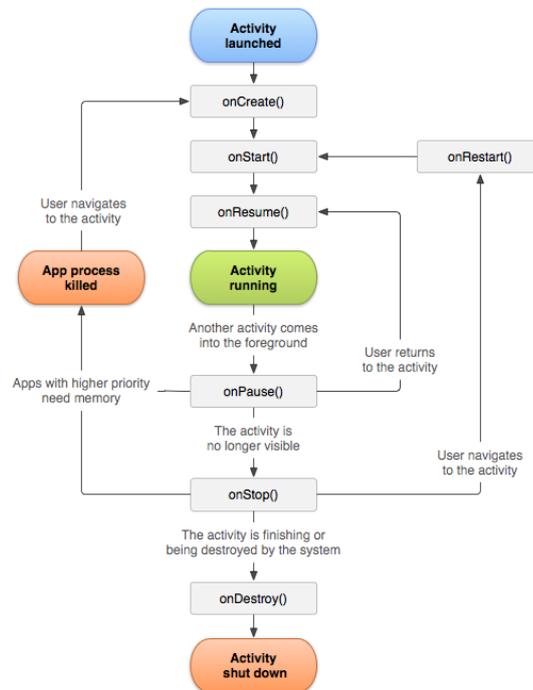


Figura 24: Ciclo de vida de una Activity

Fuente web: http://developer.android.com/images/activity_lifecycle.png

Las actividades tienen un ciclo de vida, es decir, pasan por diferentes estados desde que se inician hasta que se destruyen. Sus 3 posibles estados son:

Estado	Descripción
Activo	Ocurre cuando la actividad está en ejecución, es decir, es la tarea principal.
Pausado	La actividad se encuentra semi-suspendida, es decir, aún se está ejecutando y es visible, pero no es la tarea principal. Se debe guardar la información en este estado para prevenir una posible pérdida de datos en caso de que el sistema decida prescindir de ella para liberar memoria.
Parado	La actividad está detenida, no es visible al usuario y el sistema puede liberar memoria. En caso de necesitarla de nuevo, será reiniciada desde el principio.

Tabla 2: Estados de una Activity

Fuente web: <http://developer.android.com/intl/es/guide/components/activities.html>

También hay que tomar en cuenta qué métodos son importantes en cada uno de ellos. Aquí están los métodos más importantes de una actividad:

Método	Descripción
onCreate(Bundle savedInstanceState)	Crea la actividad, recibe un parámetro de tipo Bundle, que contiene el estado anterior de la actividad, para preservar la información que hubiera, en caso de que hubiera sido suspendida, aunque también puede iniciarse con un null si la información no es necesaria o no existe.
onRestart()	Reinicia una actividad tras haber sido parada (si continúa en la pila de tareas). Se inicia desde cero.
onStart()	Inmediatamente después de onCreate(Bundle savedInstanceState), o de onRestart() según corresponda. Muestra al usuario la actividad. Si ésta va a estar en un primer plano, el siguiente método debe ser onResume(). Si por el contrario se desarrolla por debajo, el método siguiente será onStop().
onResume()	Establece el inicio de interactividad entre el usuario y la aplicación. Solo se ejecuta cuando la actividad está en primer plano. Si necesita información previa, el método onRestoreInstanceState() aportará la situación en que estaba la actividad al llamar al onResume(). También puede guardar el estado con onSaveInstanceState().
onPause()	Se ejecuta cuando una actividad va a dejar de estar en primer plano, para dar paso a otra. Guarda la información, para poder restaurar cuando vuelva a estar activa en el método onSaveInstanceState(). Si la actividad vuelve a primer plano, el siguiente método será onResume(). En caso contrario, será onStop().
onStop()	La actividad pasa a segundo plano por un largo período. Como ya se ha dicho, el sistema puede liberar el espacio que ocupa, en caso de necesidad, o si la actividad lleva parada mucho tiempo.
onDestroy()	Es el método final de la vida de una actividad. Se llama cuando ésta ya no es necesaria, o cuando se ha llamado al método finish().

Tabla 3: Métodos que tiene una Activity

Fuente web: <http://developer.android.com/intl/es/guide/components/activities.html>

Además de los métodos anteriores, cabe destacar dos más, que son de vital importancia:

onSaveInstanceState(): guarda el estado de una actividad. Es muy útil cuando se quiere pausar una actividad para abrir otra.

onRestoreInstanceState(): restaura los datos guardados en onSavedInstanceState() al reiniciar una actividad.

- **Intents (Intenciones)**

Un intent representa un elemento básico para realizar alguna acción o tarea; como por ejemplo una llamada de teléfono o visualizar una página web. Un intent permite lanzar una actividad o servicio de una aplicación o de una aplicación diferente. Se declaran en el Android Manifest. xml con la etiqueta <Intentet>. **[WEB11]**

Tipos de Intents	
Explícitos	Implícitos
Se indica exactamente el componente a lanzar. La utilización típica es la de ir ejecutando los diferentes componentes internos de una aplicación.	Pueden solicitar tareas abstractas, como: tomar una foto, enviar un mensaje. Además se resuelven en tiempo de ejecución

Tabla 4: Tipos de Intents

Fuente web: <http://www.androidcurso.com/index.php/curso-android-basico/tutoriales-android-basico/34-unidad-3-actividades-e-intenciones/130-las-intenciones>

En otras palabras los implícitos no especifican el componente al que va destinado, mientras que el explícito, sí.

Los filtros de intenciones registran cuales son los componentes capaces de llevar una acción sobre un conjunto de datos. Los componentes se registran usando la etiqueta <intent-filter> del manifiesto especificando una acción, una categoría y unos datos. Las intenciones se pueden usar para moverse entre actividades.

- **Views (Vistas)**

Un view es un área rectangular en la pantalla que gestiona el tamaño, el dibujado, el cambio de foco y los gestos del área que representan.

La clase android.view.View sirve de base para todos los widgets. Entre las vistas disponibles tenemos: TextView, EditText, Button, RadioButton, Checkbox, DatePicker, TimePicker, Spinner, entre otros.

La interfaz de usuario para cada componente de la aplicación se define utilizando los objetos View y ViewGroup, donde cada grupo de vista es un contenedor invisible que organiza vistas, setContentView() permite añadir una vista a una actividad.

La plataforma Android ofrece dos métodos para diseñar la interfaz: Procedural (código), Declarativa (XML).

Un ViewGroup es un objeto de la clase android.view.ViewGroup, como su propio nombre lo indica, es un objeto especial de view cuya función es la de contener y controlar una lista de Views y de otros ViewGroups. Permiten añadir estructuras a la interfaz y acumular elementos complejos en la pantalla. [WEB12]

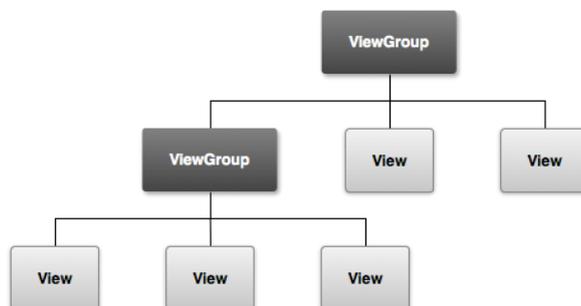


Figura 25: Árbol de jerarquía de View

Fuente web: <http://developer.android.com/images/viewgroup.png>

Los Layouts son contenedores invisibles que determinan la disposición de las vistas en la pantalla.

Layout	Descripción
LinearLayout	Dispone los hijos horizontal o verticalmente
RelativeLayout	Dispone unos elementos relativos a los otros
TableLayout	Dispone los elementos en filas y columnas
AbsoluteLayout	Dispone los elementos en coordenadas exactas
FrameLayout	Permite cambiar dinámicamente los controles en el layout

Tabla 5: Tipos de Layouts

Fuente web: <http://www.androidcurso.com/index.php/tutoriales-android/32-unidad-2-diseno-de-la-interfaz-de-usuario-vistas-y-layouts/114-layouts>

También se puede utilizar otras clases de Layouts, que se describen a continuación:
[WEB13]

- **ScrollView:** visualiza una columna de elementos; cuando éstos no caben en pantalla, permite un deslizamiento vertical.
- **HorizontalScrollView:** visualiza una fila de elementos; cuando éstos no caben en pantalla permite un deslizamiento horizontal.
- **TabHost:** proporciona una lista de ventanas seleccionables por medio de etiquetas que pueden ser pulsadas por el usuario para seleccionar la ventana que desea visualizar.
- **ListView:** visualiza una lista deslizable verticalmente de varios elementos. La utilización es algo compleja.
- **GridView:** visualiza una cuadrícula deslizable de varias filas y varias columnas.
- **ViewFlipper:** permite visualizar una lista de elementos de forma que se visualice uno cada vez. Puede ser utilizado para intercambiar los elementos cada cierto intervalo de tiempo.

• **Services (Servicios)**

Un servicio es un componente que puede realizar operaciones de larga duración en el fondo y no proporciona una interfaz de usuario. Otro componente de la aplicación puede iniciar un servicio y continuará funcionando en segundo plano, incluso si el usuario cambia a otra aplicación.

El ejemplo más simple de este concepto es el reproductor de música, que es un servicio que puede mantenerse corriendo mientras enviamos un SMS o realizamos alguna otra función en el dispositivo.

Un servicio puede tener básicamente dos formas:

- **Started:** un servicio es “started” cuando un componente de la aplicación por ejemplo una actividad se inicia con el llamado al método `startService()`. Una vez iniciado, un servicio puede ejecutarse en segundo plano de forma indefinida, incluso si el componente que se inició, se destruye.
- **Bound:** un servicio es “bound” cuando un componente de la aplicación se une mediante el llamado al método `bindService()`. Un servicio consolidado ofrece una interfaz de cliente-servidor que permite a los componentes interactuar con el servicio, enviar solicitudes, obtener resultados, e incluso hacerlo a través de los procesos de comunicación entre procesos.

El ciclo de vida de un servicio se inicia con el método `onCreate(Bundle)` y se libera con el método `onDestroy()`. Sin embargo, el desarrollo puede llevarse a cabo de dos maneras, dependiendo de cómo se lance:

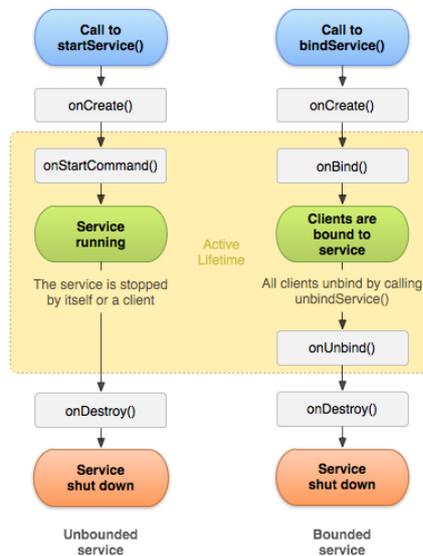


Figura 26: Ciclo de vida del Servicio

Fuente web: http://developer.android.com/images/service_lifecycle.png

Si se llama al método `startService()`, esto implicará que el servicio ejecutará todo su ciclo vital. El siguiente método tras `onCreate(Bundle)` será `onStartCommand(Intent,int,int)`. Para terminar el servicio externamente, se usa `stopService()`.

En otro caso, si el servicio se llama con `bindService()`, el usuario podría interactuar mediante la interfaz que exporta el servicio, y tras `onCreate(Bundle)` se ejecutará el método `onBind(Intent)`. En este caso, el servicio se termina llamando al método `onUnbind(Intent)`. También es posible reiniciarlo con el método `onRebind(Intent)`.

• Content Providers (Proveedores de Contenido)

Los proveedores de contenido se utilizan para gestionar y compartir bases de datos de la aplicación. Son el medio preferido de intercambio de datos a través de los límites de aplicaciones. Esto significa que se puede configurar los proveedores de contenidos propios que permitan el acceso de otras aplicaciones y hacer uso de proveedores de contenidos creados por otras personas y con ellos acceder a los datos almacenados.

Un ejemplo sencillo de esto es el proveedor de contenido que tiene Android para gestionar la información de contactos del teléfono. Cualquier aplicación con los permisos adecuados puede realizar una consulta a través de un proveedor de contenido para leer y escribir información sobre una persona en particular. **[WEB14]**

• **Broadcast Receivers (Receptores de Radiodifusión)**

(Báez, y otros, Introducción, 2012), señalan:

“También llamados como notificaciones, son los encargados de reaccionar ante los eventos ocurridos en el dispositivo, ya sean generados por el sistema o por una aplicación externa” (pág. 19)

Como por ejemplo que la pantalla se ha apagado, que la batería del teléfono está por terminarse o que una fotografía ha sido tomada. No tienen interfaz, pero pueden lanzar una activity por medio de un evento.

• **Manifest (Manifiesto)**

El manifiesto (archivo manifest.xml en cualquier proyecto Android) es un archivo en formato XML¹³, en el que se definen las características generales de una aplicación:

[WEB15]

- **El paquete:** es una cadena que describe de forma única a una aplicación, así no es posible añadir una aplicación al Play Store si ya existe otra con el mismo nombre de paquete. Si se instala en algún dispositivo una aplicación con el mismo nombre de paquete que otra ya existente, la nueva sustituirá a la anterior.
- **El nombre:** es el nombre que aparecerá en la tienda, o el nombre que enseña el instalador cuando se va a instalar la aplicación.
- **La versión:** el sistema define dos números diferentes para la versión de una aplicación: por un lado el número de compilación, que debe incrementarse en sucesivas actualizaciones (no es posible actualizar en la tienda una aplicación con un número de compilación inferior a la existente). Y por el otro el número de la versión, que es el número que se mostrará al usuario y sobre el que no hay restricciones.
- **La versión de Android a la que se dirige:** se indica la versión mínima y máxima necesaria para que la aplicación se ejecute correctamente, no siendo posible así instalar una aplicación en un dispositivo que tenga una versión de Android que no se encuentre en el rango especificado.
- **Los permisos:** lista de permisos que necesita la aplicación para ejecutarse correctamente, y que se le presentarán al usuario cuando instale la aplicación.

¹³ XML. eXtensible Markup Language – Lenguaje de Marcas eXtensible

Nota: Android no soporta aún que el usuario selecciones qué permisos desea otorgar, de entre la lista de permisos solicitados, así que éste deberá aceptar todos para poder instalar la aplicación.

- **Lista de actividades, servicios y receptores de mensajes:** aquí se enumeran todas las actividades de la aplicación, los servicios que ofrecerá y los procesos de recepción de mensajes, así como los parámetros que activarán cada uno de ellos, si los hubiera.

2.1.6 SQLITE

SQLite es una base de datos de código abierto. SQLite es compatible con las funciones de datos relacionales estándar, como la sintaxis SQL, transacciones y declaraciones preparadas. La base de datos requiere poca memoria en tiempo de ejecución aproximadamente 250 kb. **[WEB15]**

SQLite soporta los tipos de datos TEXT (similar a los String en Java), INTEGER (similar a Integer en Java) y REAL (similar a Double en Java). Si hacemos uso de cualquier otro tipo de dato, de manera automática serán convertidos para que sean compatibles con estos tres tipos de datos.

• SQLite en Android

SQLite está integrado en todos los dispositivos Android. Al momento de utilizar una base de datos SQLite en Android no se requiere un procedimiento de configuración o administración de la base de datos.

La gestión de la bases de datos la hace automáticamente Android es decir, que solo se tienen que definir las instrucciones SQL para crear y actualizar la base de datos.

Si una aplicación crea una base de datos, se almacena en:

DATA/data/APP_NAME/databases/FILENAME siendo:

- **DATA:** la carpeta de datos a través del método Environment.getDataDirectory()
- **APP_NAME:** el nombre de la aplicación
- **FILENAME:** el nombre que se da en código a la base de datos

• SQLiteOpenHelper

Esta clase permite crear, actualizar y obtener acceso a una base de datos SQLite normalmente se extiende de android.database.sqlite. Los métodos que se utilizan son:

- void **onCreate**(SQLiteDatabase db): llama al sistema, si la base de datos no existe.

- void **onUpgrade**(SQLiteDatabase db): se llama si la versión de base de datos del código ha aumentado, permitiendo actualizar el esquema de la base de datos.
- synchronized SQLiteDatabase **getReadableDatabase()**: devuelve un objeto para acceder en modo de solo lectura.
- synchronized SQLiteDatabase **getWritableDatabase()**: devuelve un objeto para acceder en modo de lectura-escritura.

Una buena práctica es crear una clase por tabla de la base de datos, para mantener el código organizado y claro.

- SQLiteDatabase

Esta clase nos permite consultar y actualizar la base de datos, para ello se necesita abrir y cerrar la base de datos.

static SQLiteDatabase **openOrCreateDatabase**(File file, SQLiteDatabase.CursorFactory factory): abre la base de datos.

void **close()**: cierra la base de datos.

Para la inserción se utiliza:

long **insert**(String table, String nullColumnHack, ContentValues values)

- **table**: nombre de la table
- **nullColumnHack**: null, o nombre de una columna si values no contiene ningún par columna-valor.
- **values**: mapa de pares nombre de columna-valor.

El método retorna el ID de la nueva fila, o -1 si ocurrió un error.

Para la actualización se tiene:

int **update** (String table, ContentValues values, String whereClause, String[] whereArgs)

- **table**: nombre de la tabla
- **values**: mapa de pares nombre de columna-valor
- **whereClause**: condición de where. Ejemplo: `_id=?`
- **whereArgs**: lista de Strings que sustituyen a `?`, o null Ejemplo: `new String[]{"1", "3", "5"}`

El método retorna el número de filas afectadas.

Para la eliminación se tiene:

int **delete** (String table, String whereClause, String[] whereArgs)

- **table**: nombre de la tabla
- **whereClause**: condición de where. Ejemplo: `_id=?`
- **whereArgs**: lista de Strings que sustituyen a `?`, o null Ejemplo: `new String[]{"1","3","5"}`

El método retorna el número de filas afectadas si `whereClause` no es null, 0 si lo es (todas serán afectadas)

Para la consulta se tiene:

Cursor **query** (String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy)

- **table**: nombre de la tabla.
- **columns**: las columnas. Null para devolver todas.
- **selection**: condición de WHERE, o null para todas las filas Ejemplo: `_id=?`.
- **selectionArgs**: lista de Strings que sustituyen a `?`, o null. Ejemplo: `new String[]{"1","3","5"}`.
- **groupBy**: argumentos de group by, o null si no aplica.
- **having**: argumentos de having, o null si no aplica.
- **orderBy**: argumentos de order by, o null si no aplica.

El método retorna un cursor posicionado en la primera entrada.

Mediante la utilización de comandos libres existen dos funciones muy útiles para enviar consultas y comandos directamente en SQL.

Cursor **rawQuery**(String sql, String[] selectionArgs)

- **sql**: consulta en SQL.
- **selectionArgs**: array de argumentos a sustituir por los `?`.

El método retorna un cursor posicionado en la primera entrada.

void **execSQL**(String sql)

- **sql**: comando en SQL que no devuelva resultados.

2.1.7 COMUNIDAD DE DESARROLLADORES ANDROID

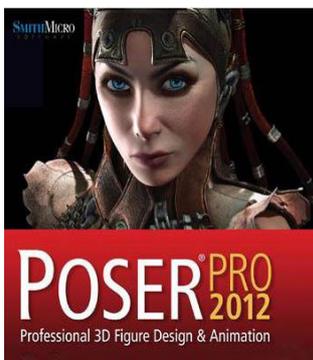
En todo el mundo existe una gran comunidad para desarrolladores, que hacen más sencillo buscar soluciones a los problemas y compartir ideas y experiencias.

- **Stack Overflow** (<http://www.stackoverflow.com>), un sitio donde se encuentra preguntas y respuestas. Si existe alguna pregunta sobre Android, es posible que ya se haya planteado la misma pregunta y que alguien ha manifestado la respuesta. Los desarrolladores pueden votar por la mejor respuesta, declarando así una respuesta de confianza.
- **Google Android Training** (<http://developer.android.com/training/index.html>), este sitio contienen un gran número de clases muy útiles que están agrupadas por temas. Las clases contienen fragmentos de código para los desarrolladores que dominan fundamentos básicos.
- **Android Discuss** (<https://groups.google.com/forum/#!forum/android-discuss>), es un grupo de discusión de Google, donde se puede tratar de cualquier tema sobre Android. Es un buen lugar para despejar dudas y aprender nuevas cosas.

2.2 SOFTWARE POSER

La mayoría de la información fue extraída de la documentación que se obtuvo al momento de instalar el programa de la siguiente ruta → C:\Program Files\Smith Micro\Poser Pro 2012\Poser Pro Reference Manual.pdf

2.2.1 INTRODUCCIÓN A POSER



Poser es una solución completa de animación en el diseño gráfico 3D¹⁴. Tanto los artistas como los animadores pueden crear escenas en 3D a partir de una variada colección de modelos humanos y animales listos para el uso de 3D de forma rápida y sencilla, utilizando la interfaz intuitiva de Poser. Contiene una biblioteca de contenido muy variada ahorrando así tiempo y recursos de desarrollo. **[WEB17]**

Poser es sin duda el estándar en la creación de figuras 3D de todo tipo y animaciones, similar a un estudio de fotografía digital que permite cargar actores, props denominados a los distintos objetos que se pueden crear, incluso escenarios completos con los que pueden interactuar los actores, también permite configurar luces, cámaras y demás.

¹⁴ 3D. Tres dimensiones. Largo, ancho y profundidad de una imagen.

• Características de Poser

- Está optimizado para sistemas de 32 y 64 bits para una mejor utilización de la memoria y de varios subprocesos, mejorando el rendimiento general y reduciendo los tiempos de procesamiento, con la renderización de fondo para que se siga trabajando con otros proyectos.
- OpenGL¹⁵ aprovecha el hardware de la tarjeta gráfica para proporcionar iluminaciones, sombras y colores realistas.
- Posee un gestor de colas que permite distribuir y administrar las representaciones a través de una cola de procesamiento.
- Es compatible con salida HDRI¹⁶ para producir imágenes con la gama completa de sombras, colores, contrastes y resaltar intensidades.
- Permite compartir el contenido final a través de la red social Facebook.
- Los trabajos realizados se guardan con un formato nativo de Poser .pz3 que ha estado durante más de 15 años.
- Incluye la selección múltiple de arrastrar y soltar desde y dentro de la biblioteca de contenido, menús contextuales ampliados para un mejor flujo de trabajo, los nuevos modos de la cámara para centrarse en el área seleccionada.
- Cuenta con una librería básica de figuras humanas, robóticas, cartoons o animales integrada con el propio programa, pero se le pueden añadir infinitas que se encuentran por toda la red.
- También incluye poses, pelo, objetos, texturas, gestos de mano o expresiones faciales.
- No tiene versión en español.

2.2.2 POSE ROOM

Pose Room es el punto focal de gran parte de actividad de Poser. Aquí es donde se agregan figuras y objetos a la escena deseada para realizar su presentación y animación. Aparece con una resolución de 1920x1200.

El número de paletas y opciones que se ve depende de la resolución de pantalla. A mayor resolución, los espacios de trabajo predeterminados muestran más paletas, mientras que a resoluciones inferiores se ve menos paletas. **[WEB17]**

¹⁵ OpenGL. Open Graphics Library – Interfaz de programación de aplicaciones.

¹⁶ HDRI. High Dynamic Range Imaging – Imágenes de alto rango dinámico

- **Menu Bar**

La barra de menú se compone de una serie de menús desplegables se utilizan para acceder a la funcionalidad de Poser. Algunos menús y/o submenús duplican otros elementos de la interfaz de pantalla.



Figura 27: Poser's Menu Bar

Fuente: propia

- **Light Controls**

Los controles de luz permiten ajustar las propiedades de iluminación, tales como agregar o quitar las luces, y la especificación de colores claros y otras propiedades de la luz.

[WEB17]



Figura 28: Light Controls

Fuente: propia

- **Camera Controls**

Los controles de la cámara permiten seleccionar y mover una o más cámaras de Poser. Hay dos tipos de controles de la cámara: View (Vista) y Position (Posición).

Los controles más altos son los controles View. Al dar clic en uno de estos controles se activa la cara, mano izquierda, mano derecha o de la cámara, según el caso. También se puede desplazar por la lista de cámaras disponibles. Los controles debajo de los controles manuales se utilizan para activar la animación de cámara encendida o apagada, o para usar la cámara que gira alrededor de la escena.

Los controles de Position aparecen por debajo de estos controles y se utilizan para mover o girar la cámara seleccionada en ese momento. Los controles de posición del plano de la cámara se mueven la cámara a lo largo de los ejes X, Y, Z; mientras que el TrackBall gira la cámara seleccionada actualmente sin cambiar su ubicación en el espacio 3D.

Al mover la cámara no cambia los lugares a las poses de las figuras, objetos y otros elementos de la escena. El botón Frame cambia la vista de la cámara para ajustar alrededor del actor seleccionado. El botón Orbit hace girar la cámara alrededor del actor actualmente seleccionado. Este control se puede alternar. Al activarse, la cámara se centra en el objeto seleccionado o parte del cuerpo y se puede girar la cámara para que esté siempre en el centro. [WEB17]

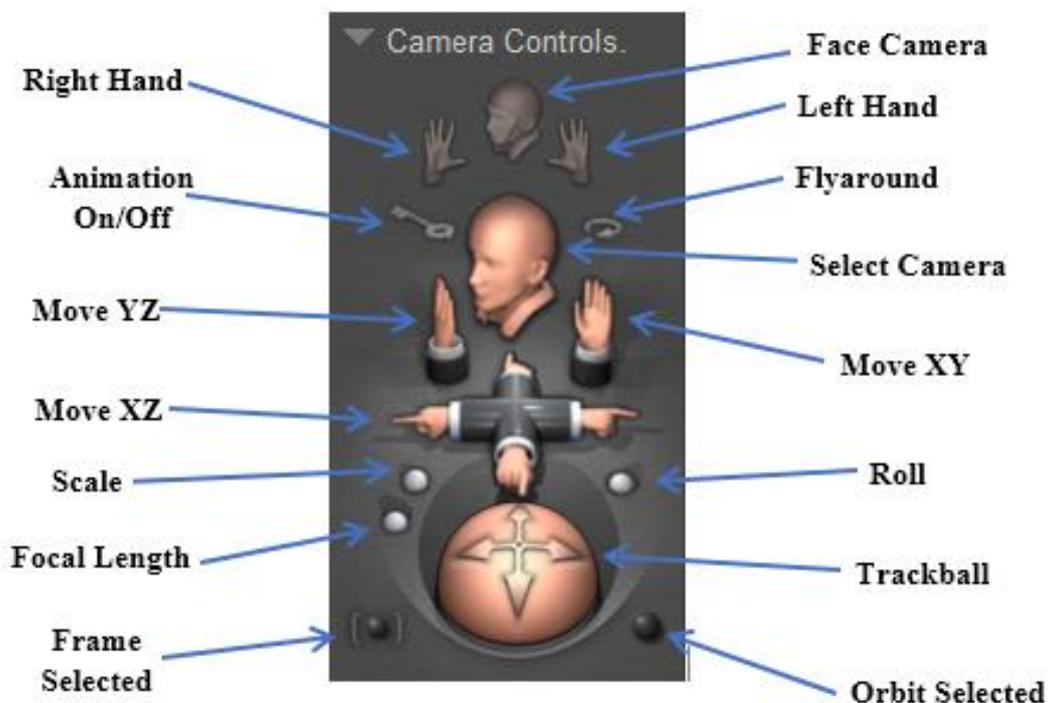


Figura 29: Camera Controls

Fuente: propia

• **Room Tabs**

Permite cambiar a las diferentes opciones como se muestra en la Figura 29.

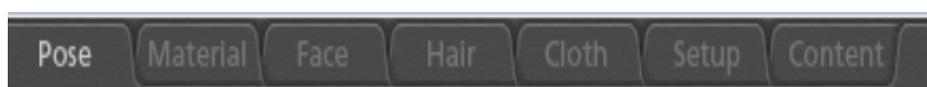


Figura 30: Room Tabs

Fuente: propia

• Editing Tools

La pantalla de las herramientas de edición contiene las herramientas que se utiliza con más frecuencia. Mediante la selección de una parte del cuerpo seguido de la herramienta de edición deseada se puede representar a los objetos de varias maneras, tales como de lado a lado o de ida y vuelta, la rotación, torsión y otros. **[WEB17]**

- **Rotate:** esta herramienta inclina una parte del cuerpo en su conjunto. Trabaja en tres dimensiones, lo que permite realizar un gran progreso muy rápidamente.
- **Twist:** permite girar una figura, parte a lo largo de su eje más largo (longitud). Por ejemplo el brazo puede girar casi 180 grados.
- **Translate/Pull:** desplaza la figura, parte seleccionada junto a la visión de la cámara. Se puede realizar en los ejes X, Y, Z dependiendo de la posición de la cámara respecto a la figura.
- **Translate In/Out:** esta herramienta desplaza la figura, parte seleccionada en el eje Z de la cámara (dentro y fuera). Se puede realizar a lo largo de los ejes X, Y, Z de la figura.
- **Scale:** esta herramienta permite ampliar o reducir la cifra de la parte seleccionada en los ejes X, Y de la cámara.
- **Taper:** esta herramienta permite graduar la figura, pieza seleccionada a lo largo de la cámara de los ejes X e Y. Estos ejes pueden variar dependiendo de la posición de la cámara.
- **Chain Break:** esta herramienta termina la cadena de las partes afectadas por la herramienta Translate tools. Los saltos de cadena en diferentes partes del cuerpo pueden crear efectos interesantes. Por ejemplo, para evitar que el pecho se mueva, se aplica chain break a la misma.
- **Color:** permite cambiar de color la superficie de un grupo de materiales.
- **Grouping tool:** esta herramienta abre la paleta Group Editor.
- **View Magnifier:** permite ampliar y reducir desde y hacia las áreas deseadas de la ventana del documento, sin alterar la posición de la cámara seleccionada.
- **Morphing Tool:** abre la paleta Morph Editor. La herramienta Morphing Tool tiene dos modos de funcionamiento: combinar y crear, que se puede acceder y controlar a través del Morph Editor. El modo combinar permite esculpir cualquier superficie en la figura. El modo crear permite manipular los vértices en la malla del modelo.

- **Direct Manipilation:** permite ajustar directamente los parámetros traslación, escala y rotación. También se puede utilizar esta herramienta para ajustar las zonas de atenuación a la hora de establecer los parámetros comunes de las poses para una figura.

Cada herramienta de edición es un poco diferente, pero su uso básico es el siguiente:

- ✓ Se selecciona la parte del cuerpo que se desea en la escena haciendo clic en el actor.
- ✓ Se selecciona la herramienta de edición deseada. Sólo se puede tener una herramienta de edición seleccionada a la vez, la herramienta seleccionada aparece de color amarillo en la interfaz de Poser.
- ✓ Se da un clic y se arrastra para llevar a cabo la función de la herramienta seleccionada.



Figura 31: Editing Tools

Fuente: propia

- Document Window

La ventana de documento es un espacio de trabajo donde se puede ver y crear poses de una figura e interactuar directamente con la escena. Cada vista de la ventana de documento se muestra a través de una cámara virtual, lo que significa que se puede ver cada escena de múltiples ángulos de cámara, ya sea de uno a uno o de a cuatro ángulos a la vez.

Se puede colocar las cámaras para ver la escena desde cualquier ángulo o la distancia y se puede cambiar el tamaño de la ventana de documento para satisfacer las necesidades. Además, la ventana de documento tiene numerosos controles alrededor del borde que se utilizan para cambiar la apariencia de los elementos de la escena. **[WEB17]**

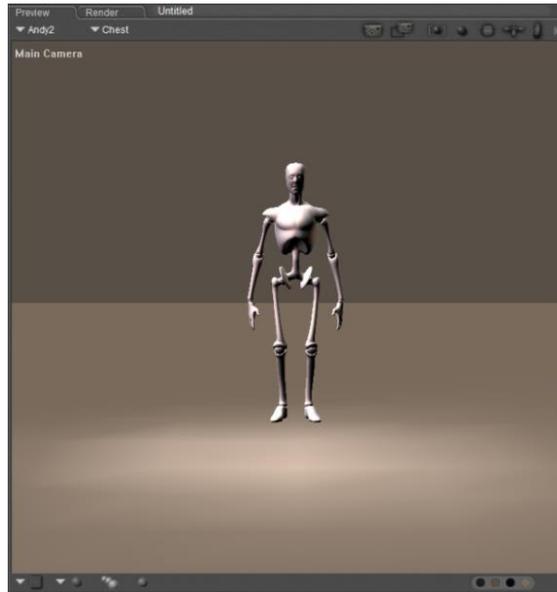


Figura 32: Document Window

Fuente: propia

• Display Controls

Los controles de pantalla permiten seleccionar el modo de vista previa que se desea para toda la escena, una figura o prop¹⁷, elemento específico de una figura seleccionada.

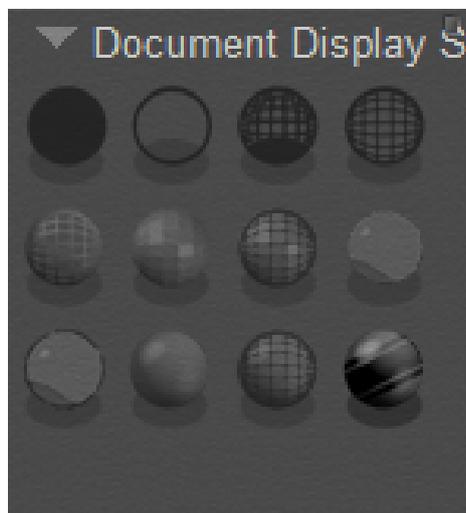


Figura 33: Display Controls

Fuente: propia

¹⁷ Prop. Objeto móvil utilizado en el escenario por los actores durante una producción de la pantalla

- Parameters/Properties Palette

Poser ofrece una solución fácil de usar parámetros y la paleta de propiedades que permite acceder a las propiedades y los parámetros de cualquier objeto de la escena.

[WEB17]

- Parameters Palette (Paleta de Parámetros)

Contiene todos los parámetros de configuración para la transformación de un objeto, cada tipo de objeto tiene parámetros especializados. Además, los objetos individuales pueden tener parámetros únicos para ese objeto.

- Properties Palette (Paleta de Propiedades)

Contiene todas las propiedades de un objeto, además de ciertas propiedades universales.



Figura 34: Parameters Palette

Fuente: propia

• UI dots

Son puntos de memoria que permite guardar posturas, ubicación de cámaras y la interfaz de usuario. En otras palabras tiene la opción de personalizar el espacio de trabajo.

[WEB17]



Figura 35: UI Dots

Fuente: propia

• Animation Controls

Los controles de animación permiten pre visualizar las animaciones y configurar fotogramas clave¹⁸. Al momento de aplastar el botón Reproducir se reproduce la animación con los fotogramas clave. **[WEB17]**

Esta paleta contiene las herramientas más utilizadas en la creación de animaciones. Permiten agregar y editar fotogramas clave y una vista previa de la animación. Si los controles de animación no aparecen en la pantalla se selecciona Window → Animation Controls y se la puede colocar donde sea más conveniente (por lo general en la parte superior o inferior de la ventana del documento). Los controles de animación se muestran de la siguiente manera:

- Timeline/Frame Indicator

La parte central de los controles de animación muestra la línea de tiempo, lo que representa el tiempo total de la animación¹⁹. Esta línea de tiempo contiene un puntero (Scrubber) que representa la ubicación del cuadro actual con relación a toda la animación, que es visible en la ventana del documento. El Scrubber o depurador es comúnmente muy utilizado para encontrar rápidamente puntos específicos dentro de las animaciones.

En la parte superior del Scrubber se encuentra el Frame Indicator, que indica tanto el número de fotogramas de la animación y su posición actual dentro de esa línea de tiempo, el campo de la parte izquierda indica la posición y el campo de la derecha es la longitud de la animación. Por ejemplo si están presentes los números 60 y 120 significa que se está viendo el fotograma²⁰ 60 de una animación que contiene 120 fotogramas. **[WEB17]**

- Play Controls

Los controles de reproducción permiten obtener una vista previa de la animación mediante controles VCR (Video Cassette Recorder). **[WEB17]**

Los controles de reproducción son los siguientes:

- ✓ **First Frame:** se mueve al primer fotograma
- ✓ **End Frame:** se mueve al último fotograma.
- ✓ **Stop:** detiene la reproducción de la animación.

¹⁸ Fotogramas clave. Son fotogramas con un contenido específico.

¹⁹ Animación. Es una secuencia de fotogramas reproducidos a una velocidad alta.

²⁰ Fotograma. Representa el contenido de una animación o video en un instante de tiempo.

- ✓ **Play/Pause:** reproduce la animación a velocidad normal, incluyendo sonidos de fondo, mientras que la animación se está reproduciendo, éste botón se convierte en el botón pausa que detiene la animación en el cuadro actual.
- ✓ **Step Backward:** se mueve a la imagen anterior de la actual.
- ✓ **Step Forward:** se mueve a la imagen siguiente de la actual.
- ✓ **Loop:** hace que la animación se repita continuamente durante la reproducción.

- Keyframe Controls

En la mayoría de los casos el motor de animación registra automáticamente los fotogramas clave al crear animaciones; sin embargo a veces se tendrá que añadir manualmente. Es importante hacer un seguimiento de la situación de la animación, ya que se podría crear una serie de fotogramas clave sin darse cuenta. Los controles de fotogramas clave permiten de una manera rápida y fácil de añadir, eliminar, tener una vista previa de los fotogramas clave. **[WEB17]**

Los controles clave son los siguientes:

- ✓ **Previous Keyframe:** vuelve al fotograma clave anterior.
- ✓ **Next Keyframe:** avanza al siguiente fotograma clave.
- ✓ **Edit Keyframes:** edita fotogramas clave.
- ✓ **Add Keyframes:** añade un fotograma clave en el número de fotograma actual.
- ✓ **Remove Keyframes:** elimina el fotograma clave seleccionado.
- ✓ **Skip Frames:** este botón se activa para saltar fotogramas durante la reproducción. Esto ayuda a conservar los recursos del sistema durante la vista previa de la animación antes de hacer el render.

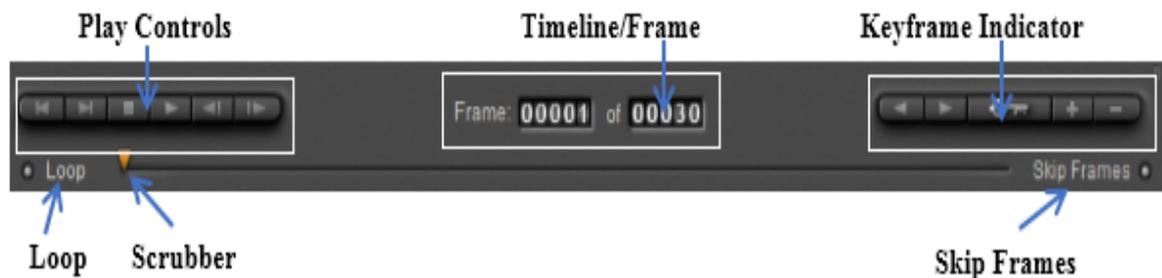


Figura 36: Animation Controls

Fuente: propia

• Library Palette

La paleta de biblioteca consta de:

- Library Tab

Contiene una biblioteca de categorías. Estos iconos de categorías permiten acceder y buscar a través de todos los contenidos disponibles. Las categorías pueden ser: figuras, accesorios y otros, y subcategorías animales, dibujos, entre otros., para organizar el contenido.

- Search Tab

Se puede utilizar para realizar búsquedas de palabras clave en una o todas las bibliotecas en tiempo de ejecución.

- Favorites Tab

Proporciona un acceso rápido al contenido de Poser favorito.

2.2.3 POSER LIBRARY

Proporciona un acceso a todo el contenido que incluye Poser como figuras, objetos, luces, cámaras y otros. Posee tres pestañas: la pestaña biblioteca que proporciona acceso a las categorías de contenido Poser, la pestaña buscar que permite buscar las carpetas de contenido para archivos específicos y la pestaña favoritos, donde se puede acceder fácilmente al contenido de uso frecuente o preferido.

Cada categoría se divide en subcategorías, se puede crear y eliminar categorías y subcategorías. También se puede agregar y eliminar contenido, incluyendo las bibliotecas de contenido existentes. **[WEB17]**

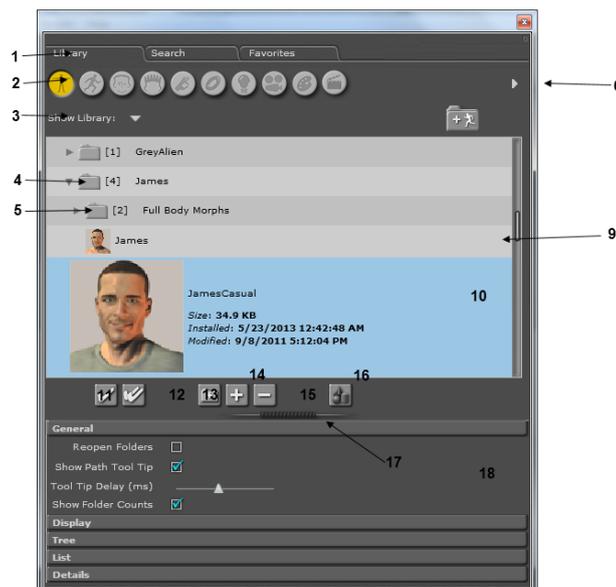


Figura 37: Library Window

Fuente: propia

- 1. Library Tabs:** permite cambiar entre las tres vistas como son: Library (Biblioteca), Search (Buscar) y Favorites (Favoritos).
- 2. Category Icons:** permite seleccionar una de las diez categorías de la biblioteca, de izquierda a derecha se tiene: Figures (figuras), Poses (poses), Hair (pelo), Hands (manos), Props (objetos), Lights (luces), Cameras (cámaras), Materials (materiales) and Scenes (escenas).
- 3. Library Selection:** selector que muestra todas las carpetas en ejecución, o muestra los contenidos de una carpeta seleccionada en tiempo de ejecución.
- 4. Expanded Folder:** carpeta expandida, con la flecha abajo contrae el contenido de la carpeta. El número que aparece entre paréntesis al comienzo del nombre de la carpeta indica el número de elementos que se encuentran dentro de esa carpeta.
- 5. Collapsed Folder:** carpeta contraída, con la flecha hacia la derecha amplía el contenido de la carpeta. El número que aparece entre paréntesis al comienzo del nombre de la carpeta indica el número de elementos que se encuentran dentro de esa carpeta.
- 6. Category Selection:** permite seleccionar una categoría de la biblioteca por su nombre.
- 7. Remove Library:** esta opción aparece al momento de eliminar una librería.
- 8. Add Library:** permite seleccionar las carpetas de la biblioteca adicionales almacenadas en una carpeta específica.
- 9. Item List Panel:** al ser seleccionado se muestra resaltado con el color azul.
- 10. Extended Details Panel:** muestra en miniatura el nombre del archivo, tamaño, fecha de instalación y otros detalles sobre el elemento seleccionado en la biblioteca.
- 11. Single Checkmark:** sustituye el elemento seleccionado actual en la escena con el elemento que se resalta en la biblioteca.
- 12. Double Checkmark:** agrega el elemento seleccionado a la escena.
- 13. Add Folder:** crea una nueva carpeta o subcarpeta debajo de la carpeta de la biblioteca seleccionada.
- 14. Save to Library:** agrega el elemento seleccionado en ese momento en la escena de la biblioteca seleccionada.
- 15. Delete from Library:** borra el elemento de contenido de contenido de la biblioteca que está seleccionado actualmente. Elimina por completo de la biblioteca y no de la escena.

16. Add to Favorites: añade el elemento que está resaltado actualmente en la biblioteca de favoritos.

17. Display Options Widget: permite expandir o contraer las opciones de visualización en la parte inferior de la paleta de la biblioteca.

18. Display Options: sirve para configurar las opciones generales de visualización.

19. Breadcrumb Trail: muestra la ruta de la biblioteca a la carpeta o el elemento seleccionado en ese momento.

20. Library Menu: solo aparece en la biblioteca de 32 bits.

• **Rendering Animations**

Se puede acceder a la ventana Movie Settings seleccionando Animation → Make Movie. Para utilizar la ventana Movie Settings se puede seleccionar el formato de vídeo, las opciones disponibles son: **[WEB17]**

- **Image Files:** crea una secuencia de imágenes, que es una serie de imágenes fijas, una para cada fotograma de la animación. Se recomienda utilizar esta opción para obtener los máximos resultados de valor y de calidad óptima.

- **AVI (Windows) / QuickTime (Macintosh):** estos son los formatos de vídeo por defecto de Windows / Macintosh. Se recomienda utilizar éstos formatos sólo para fines de pre visualización mientras se trabaja en la animación, ya que proporciona los más rápidos y posibles resultados proporcionados.

- **Flash:** es un formato popular para animaciones publicadas en los sitios web.

CAPÍTULO III

DISEÑO Y DESARROLLO DE LA APLICACIÓN

3.1 METODOLOGÍA DE DESARROLLO ASD

3.1.1 INTRODUCCIÓN

Adaptive Software Development (Desarrollo de Software Adaptable), ésta técnica fue desarrollada por Jim Highsmith y Sam Bayer a comienzos de 1990. El funcionamiento es cíclico, así se puede dar en cada iteración cambios e incluso errores.

Es una metodología abierta, es decir, no tiene muchas ataduras ni reglas a seguir, referente a la documentación no existe técnicas o herramientas para abordar estos puntos.

3.1.2 CARACTERÍSTICAS

- Es iterativo.
- Orientado en los componentes de software.
- Tolerante a cambios.
- Guiado por los riesgos.
- La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

3.1.3 CICLO DE VIDA

ASD utiliza un cambio orientado hacia el ciclo de vida, tiene 3 componentes que son; especular, colaborar y aprender.

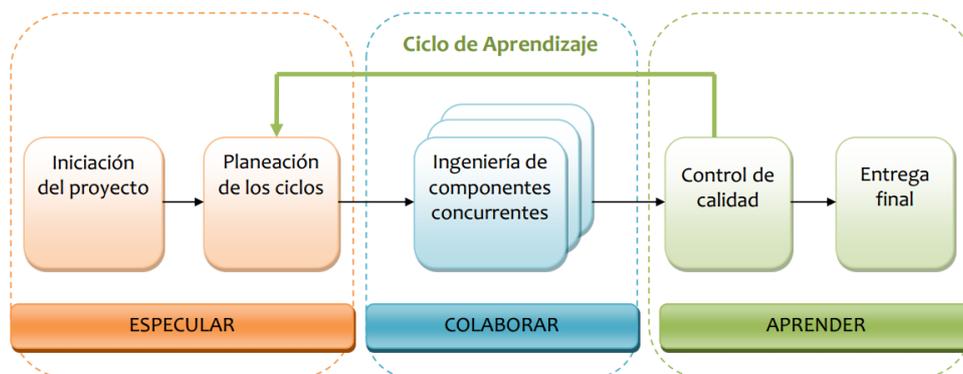


Figura 38: Ciclo de vida de la metodología ASD

Fuente web: <http://es.scribd.com/doc/141283216/Asd#download>

- **Especular:** fase de iniciación donde se establecen los principales objetivos y metas del proyecto en su conjunto, además las limitaciones o zonas de riesgo con las cuales operará el proyecto. Se realiza una planificación tentativa del proyecto con estimaciones de tiempo sabiendo que pueden sufrir desviaciones.
- **Colaborar:** en ésta fase se toma el rumbo planteado en la fase de especular, se construye la funcionalidad del proyecto. También se puede explorar nuevas alternativas que cambien el rumbo del proyecto, por eso ASD está dentro de metodologías ágiles. Aquí no se propone técnicas ni tareas al momento de la construcción.
- **Aprender:** la última fase consiste en capturar todo lo que se ha aprendido, tanto positivo como negativo. Se revisa la calidad y se entrega al cliente.

3.1.4 VENTAJAS Y DESVENTAJAS

• Ventajas

- La tercera fase, revisión de los componentes, se aprende de los errores, esto sirve para volver a iniciar el ciclo de desarrollo.
- Utiliza información disponible acerca de cambios para mejorar el comportamiento del software.
- Difunde la colaboración de distintas personas.

• Desventajas

- Al ser una metodología ágil implica no realizar procesos que son requeridos en las metodologías tradicionales.
- El ciclo entre el aprendizaje y la especulación es bueno, pero si los errores o cambios no son detectados puede afectar tanto a la calidad del producto como al costo total.

3.2 FASES DE LA METODOLOGÍA

3.2.1 FASE ESPECULAR

Con esta fase se pretende realizar el proceso de desarrollo adaptable para la realización de la aplicación, se establecen ciclos o iteraciones. Esto es una planificación tentativa del proyecto indicando su fecha y actividad correspondiente.

3.2.1.1 MISIÓN

Brindar información sobre actuaciones de emergencia en otras palabras de primeros auxilios a los usuarios, es muy importante que las personas tengan fundamentos aunque éstos sean básicos, porque no se sabe cuándo puede producirse un accidente.

3.2.1.2 LIMITACIONES

- Puede ser instalada a partir de la versión 2.3 Gingerbread en adelante.
- Para realizar las llamadas el usuario tiene que tener saldo en el dispositivo.
- Si se desea instalar en las tablets éstas deben tener la tarjeta SIM²¹ incorporada.
- Poseer espacio disponible para su instalación.
- Solo se pueden ingresar números de teléfonos celulares.

3.2.1.3 CICLO 1 (PLAN DE CICLOS)

Iteración	Fecha	Actividad
1	2013-10-12	Corresponde a la especulación, se determina el número de ciclos y sus actividades, ésta información era tentativa debido a la poca información que se tenía.
2	2013-10-20	Conforme se obtuvo más información se redefinieron los nombres de los ciclos 2, 3, 4 y 5. Esto alteró un poco el cronograma pero no ocasionó ningún problema en el desarrollo de la aplicación.
3	2013-11-13	Se encontró un problema en las actividades del ciclo 3, ya que no se encontraban bien definidas.
4	2013-12-12	Se corrigió algunas actividades en los ciclos de análisis, diseño e implementación.
5	2014-01-10	Se encontró problemas en el ciclo de pruebas y documentación en cuanto a la reproducción de los videos.

Tabla 6: Ciclo de planificación

Fuente: propia

²¹ SIM. Subscriber Identity Module – Módulo de Identificación de Abonado.

3.2.1.4 CICLO 2 (ANÁLISIS)

Iteración	Fecha	Actividad
1	2013-10-01	Se crea la primera versión de la aplicación
2	2013-10-06	Para complementar el análisis se crearon diagramas de caso de uso.
3	2013-10-15	Se corrigieron requerimientos para el segundo módulo.
4	2013-10-20	Los diagramas de caso de uso para los módulos son los mismos con excepción del segundo módulo.
5	2013-10-22	Se corrigió la documentación de los casos de uso.
6	2013-10-26	Se corrigió los requerimientos de los módulos que presentan las animaciones.

Tabla 7: Ciclo de análisis
Fuente: propia

3.2.1.5 CICLO 3 (DISEÑO)

Iteración	Fecha	Actividad
1	2013-11-05	Se define las diferentes clases que formarán parte de la aplicación.
2	2013-11-10	Se agregó el modelo de datos y su documentación para el segundo módulo.
3	2013-11-16	Se definió la arquitectura funcional y tecnológica.
4	2013-11-21	Se corrigió errores en cuanto a la arquitectura.
5	2013-11-23	Se corrigió el modelo de datos.
6	2013-11-26	Se agregaron otras clases para el funcionamiento de la aplicación.
7	2013-11-30	Se mejoró la descripción de arquitectura y de algunos diagramas.

Tabla 8: Ciclo de diseño
Fuente: propia

3.2.1.6 CICLO 4 (IMPLEMENTACIÓN)

Iteración	Fecha	Actividad
1	2013-12-01	Se implementó las clases o actividades para cada módulo y para mostrar las animaciones.
2	2013-12-04	Se corrigió errores para la visualización de los videos.
3	2013-12-06	Se implementó la base de datos.
4	2013-12-10	Para la visualización de los videos se definió que ocupe toda la pantalla.
5	2013-12-12	Se mejoró la interfaz gráfica principal.
6	2013-12-15	Se corrigió errores en el segundo módulo para la edición de un contacto.
7	2013-12-18	Se creó validaciones de los campos del contacto para insertar solo números celulares.
8	2013-12-20	Se implementó la opción de llamar a un contacto.
9	2013-12-24	Se realizaron algunos ajustes de acuerdo a las pruebas.

Tabla 9: Ciclo de implementación

Fuente: propia

3.2.1.7 CICLO 5 (PRUEBAS Y DOCUMENTACIÓN)

Iteración	Fecha	Actividad
1	2013-12-30	Se subió el archivo .apk firmado al Play Store.
2	2014-01-01	Se instaló la aplicación en dispositivos con diferentes tamaños de pantalla.
3	2014-01-15	Se realizaron el manual de usuario y el manual de instalación y configuración.
4	2014-01-18	Se comprobó que se realicen las llamadas a los contactos almacenados.
5	2014-01-25	Se corrigieron algunos errores en los manuales y la aplicación.

Tabla 10: Ciclo de pruebas y documentación

Fuente: propia

Las iteraciones mencionadas con anterioridad no fueron realizadas en forma lineal, además se basaron de acuerdo al aprendizaje obtenido de la iteración previa. Hay que señalar que los ciclos y actividades se fueron modificando según el avance del proyecto.

3.2.2 FASE COLABORAR

Se realizan las características que se plantearon en la fase de especulación lo cual era una investigación. Se detallan mediante un plan de ciclos que fueron sometidos a varias revisiones ya que ésta metodología al ser iterativa lo permite, ayudando así a controlar los cambios que se obtenían conforme avanzó el proyecto.

A continuación se muestra más detalladamente las actividades realizadas en cada ciclo citado en la fase anterior.

3.2.2.1 ANÁLISIS

Se empezó con el ciclo de análisis, este ciclo plantea las siguientes actividades:

- Se definió que la aplicación funcionará a partir de la versión de 2.3.3 de Android, ya que la mayoría de aplicaciones están instaladas desde esta versión.
- En el análisis de requerimientos se hizo referencia a los módulos de las animaciones y al módulo información del lesionado obteniendo los siguientes requerimientos:

Requerimiento	Módulos
<ul style="list-style-type: none"> - Reproducir los videos en toda la pantalla, la orientación debe ser horizontal. - La información al usuario debe ser clara y precisa con personajes 3D virtuales. - El formato de video deber ser compatible con cualquier dispositivo móvil android. 	Valoración del lesionado, Reanimación Cardio Pulmonar, Fracturas, Quemaduras y Botiquín de emergencia.
<ul style="list-style-type: none"> - Crear una agenda de contactos. - Agregar, editar, buscar, eliminar, llamar un contacto. - Utilizar los datos (nombre, teléfono) de la agenda del dispositivo para agregar un contacto. 	Información del lesionado.

Tabla 11: Requerimientos de los módulos

Fuente: propia

- **Participantes o equipo de trabajo**

El equipo de trabajo debe ser pequeño para facilitar la comunicación, deben colaborar en todo sentido para la realización del proyecto.

Nro.	Rol	Nombre
1	Cliente	Usuario/a
2	Líderes del proyecto	Daniel Medina Ing. Carpio Pineda
3	Programador, desarrollador	Daniel Medina
4	Tester	Daniel Medina Ing. Carpio Pineda

Tabla 12: Participantes o equipo de la aplicación móvil

Fuente: propia

- **Diagramas de caso de uso**

- **Caso de uso visualizar videos**

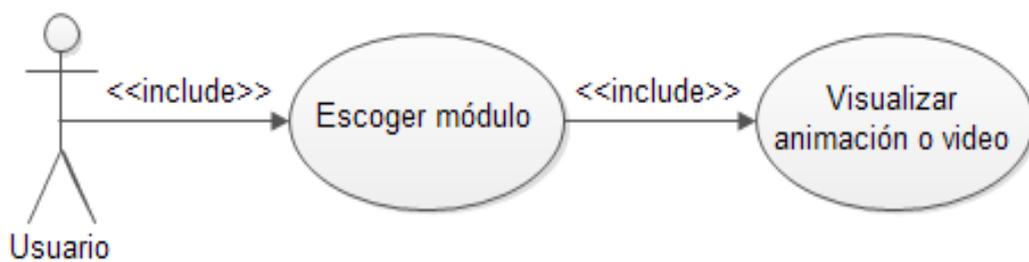


Figura 39: Proceso visualizar animaciones

Fuente: propia

• Descripción del caso de uso

Nombre:	Visualizar animaciones
Autor:	Daniel Medina
Fecha:	2013-10-21
Descripción:	Permite visualizar las animaciones o videos para los módulos valoración del lesionado, reanimación cardio pulmonar, fracturas, quemaduras y botiquín de emergencia.
Actor:	Usuario de la aplicación.
Precondiciones:	El usuario tiene que escoger el módulo de la animación que desea visualizar.
Flujo normal:	<ol style="list-style-type: none"> 1. El actor pulsa el botón ver video. 2. La aplicación busca el id del video al cual se hace referencia. 3. La aplicación ejecuta el video de forma horizontal ocupando toda la pantalla del dispositivo. 4. El actor visualiza el video que está en ejecución.
Flujo alternativo:	Ninguna
Poscondiciones:	La animación del módulo elegido se mostrará correctamente.

Tabla 13: Descripción del proceso visualizar las animaciones

Fuente: propia

• Caso de uso crear nuevo contacto



Figura 40: Proceso crear nuevo contacto

Fuente: propia

• **Descripción del caso de uso**

Nombre:	Llamar contacto
Autor:	Daniel Medina
Fecha:	2013-10-21
Descripción:	
Permite al usuario crear contactos para avisar en caso de emergencia, ya sea llenando los datos directamente o tomando el nombre y teléfono de la agenda de contactos del dispositivo.	
Actor:	
Usuario de la aplicación.	
Precondiciones:	
El usuario debe disponer de saldo para realizar la llamada. El usuario debe haber ingresado por lo menos un contacto en el caso de uso crear nuevo contacto.	
Flujo normal:	
<ol style="list-style-type: none"> 1. El actor pulsa el botón mis contactos. 2. La aplicación muestra una lista de contactos ordenados si han sido ingresados con anterioridad, caso contrario no mostrará nada, una caja de texto para buscar el contacto. 3. El actor pulsa sobre la caja de texto para buscar por nombre al contacto. 4. La aplicación devuelve las coincidencias del nombre ingresado. 5. El actor pulsa sobre el contacto encontrado y escoge la opción llamar. 6. La aplicación muestra un cuadro de diálogo con dos opciones. 7. El actor pulsa el botón Si para realizar la llamada o pulsa sobre No para cancelar la llamada. 8. La aplicación según lo escogido por el actor realiza o no la llamada. 	
Flujo alternativo:	
La aplicación no realizará la llamada por no disponer de saldo o haber cancelado la llamada.	
Poscondiciones:	
Llamando al número de teléfono.	

Tabla 14: Descripción del proceso crear nuevo contacto

Fuente: propia

• **Caso de uso llamar contacto**

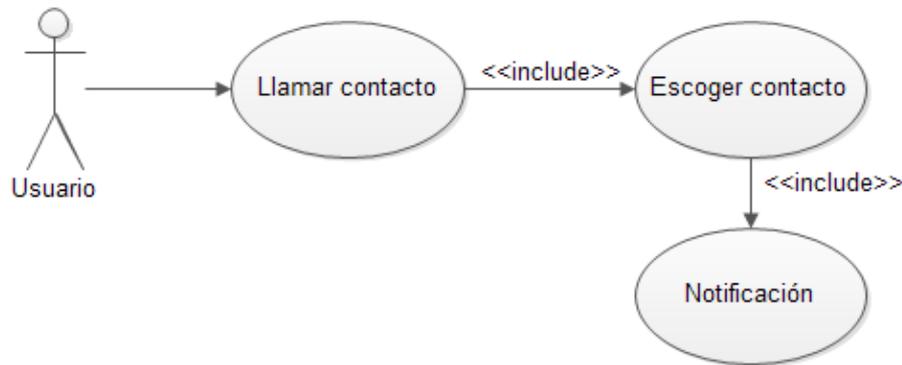


Figura 41: Proceso llamar contacto

Fuente: propia

• **Descripción del caso de uso**

Nombre:	Crear nuevo contacto
Autor:	Daniel Medina
Fecha:	2013-10-22
Descripción:	Permite al usuario realizar una llamada telefónica a un determinado contacto en caso de alguna emergencia.
Actor:	Usuario de la aplicación.
Precondiciones:	El usuario tiene que escoger el módulo de información del accidentado.
Flujo normal:	<ol style="list-style-type: none"> 1. El actor pulsa el botón mis contactos. 2. La aplicación muestra una lista de contactos ordenados si han sido ingresados con anterioridad, caso contrario no mostrará nada. 3. El actor pulsa el botón añadir nuevo contacto. 4. La aplicación proporciona dos cajas de texto para introducir el nombre, el número de teléfono, una lista de opciones para escoger el tipo de relación con el contacto que se va a ingresar, un botón añadir contacto de la agenda. 5. El actor pulsa el botón añadir contactos de la agenda. 6. La aplicación muestra todos los contactos de la agenda del dispositivo. 7. El actor pulsa en el contacto elegido, escoge el número de teléfono y pulsa sobre aceptar. 8. La aplicación llena los campos de texto nombre y teléfono con los datos del contacto elegido por el actor.

9. El actor escoge el tipo de relación.
10. El sistema verifica la validez de los datos ingresados.
11. El actor pulsa guardar.
12. La aplicación crea el contacto con los datos y lo almacena.

Flujo alternativo:

La aplicación no ingresará el contacto si todos los datos no han sido ingresados y validados correctamente.

Poscondiciones:

El contacto ha sido ingresado.

Tabla 15: Descripción del proceso llamar contacto

Fuente: propia

• **Caso de uso editar contacto**

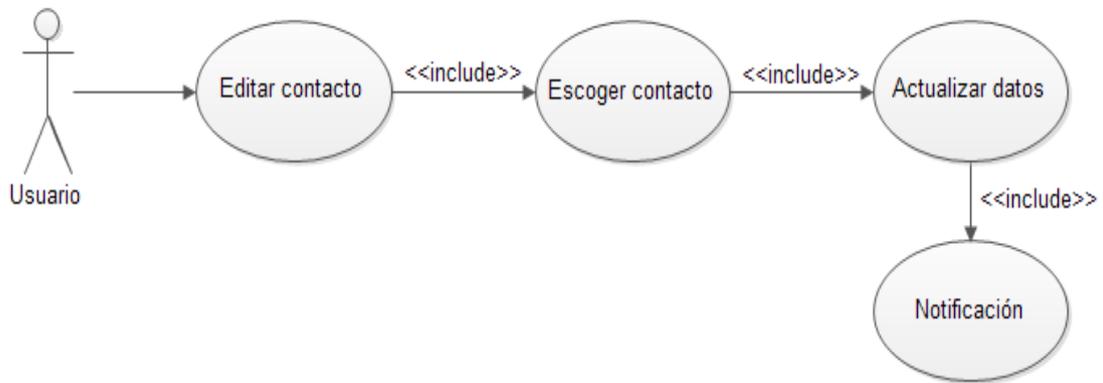


Figura 42: Proceso editar contacto

Fuente: propia

• **Descripción del caso de uso**

Nombre:	Editar contacto
Autor:	Daniel Medina
Fecha:	2013-10-22
Descripción:	Permite al usuario modificar los datos de un contacto seleccionado.
Actor:	Usuario de la aplicación.
Precondiciones:	El usuario debe buscar el contacto para actualizar los datos.

Flujo normal:

1. El actor pulsa el botón mis contactos.
2. La aplicación muestra una lista de contactos ordenados si han sido ingresados con anterioridad, caso contrario no mostrará nada, una caja de texto para buscar el contacto.
3. El actor pulsa sobre la caja de texto para buscar por nombre al contacto.
4. La aplicación devuelve las coincidencias del nombre ingresado.
5. El actor pulsa sobre el contacto encontrado y escoge la opción de editar.
6. La aplicación extrae los valores de la base de datos del contacto seleccionado.
7. El actor modifica los valores del contacto.
8. La aplicación actualiza los datos del contacto.

Flujo alternativo:

La aplicación no actualizará el contacto si todos los datos no han sido ingresados y validados correctamente.

Poscondiciones:

Los datos del contacto han sido actualizados.

Tabla 16: Descripción del proceso editar contacto

Fuente: propia

• **Caso de uso eliminar contacto**

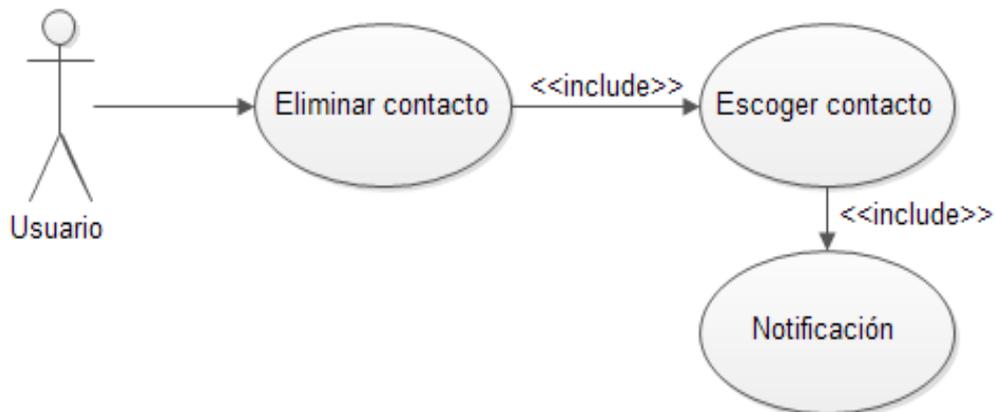


Figura 43: Proceso eliminar contacto

Fuente: propia

• **Descripción del caso de uso**

Nombre:	Eliminar contacto
Autor:	Daniel Medina
Fecha:	2013-10-22
Descripción:	Permite al usuario eliminar un contacto seleccionado
Actor:	Usuario de la aplicación.
Precondiciones:	El usuario debe buscar el contacto que va a ser eliminado.
Flujo normal:	<ol style="list-style-type: none"> 1. El actor pulsa el botón mis contactos. 2. La aplicación muestra una lista de contactos ordenados si han sido ingresados con anterioridad, caso contrario no mostrará nada, una caja de texto para buscar el contacto. 3. El actor pulsa sobre la caja de texto para buscar por nombre al contacto. 4. La aplicación devuelve las coincidencias del nombre ingresado. 5. El actor pulsa sobre el contacto encontrado, escoge la opción de eliminar y pulsa sobre aceptar. 6. La aplicación extrae los valores de la base de datos del contacto seleccionado y elimina el contacto de la base de datos.
Flujo alternativo:	El usuario al pulsar aceptar elimina el contacto, al pulsar cancelar no lo elimina.
Poscondiciones:	El contacto ha sido eliminado.

Tabla 17: Descripción del proceso eliminar contacto

Fuente: propia

3.2.2.2 DISEÑO

Para el desarrollo de una aplicación, sistema, proyecto, entre otros; es muy importante realizar un diseño previo a la implementación. Es necesario realizar diagramas que expliquen el funcionamiento de lo que se propone llegar a realizar.

• **Diagrama de datos**

Con los requisitos definidos en la etapa de análisis se diseñó un modelo de datos para el módulo de información del accidentado, el cual pedía permitir la creación de una agenda de contactos.

Para poder visualizar la base de datos creada se puede utilizar el software sqlite admin, obtenido de: <http://sqliteadmin.software.informer.com/0.8b/>.

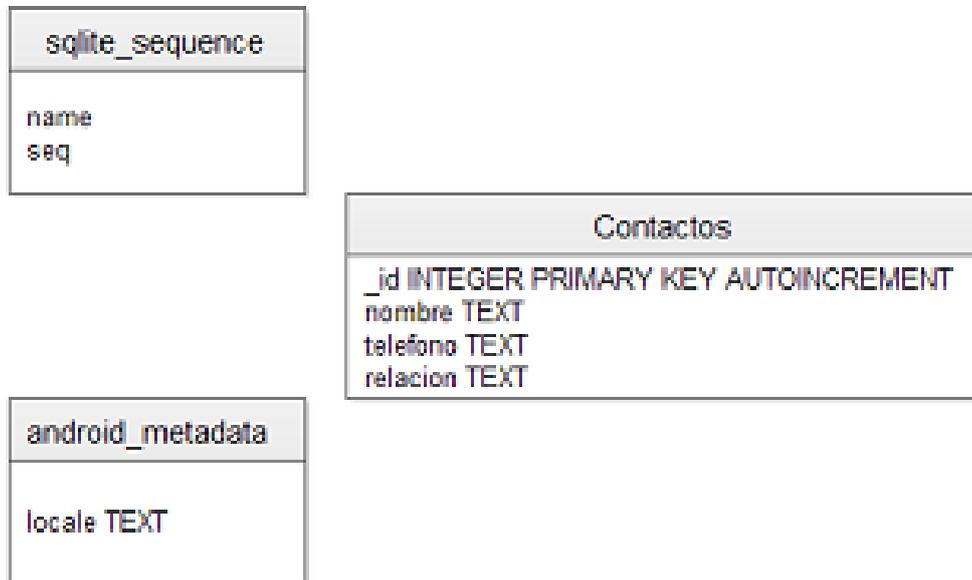


Figura 44: Diagrama de datos módulo información del lesionado

Fuente: propia

• Diagrama arquitectónico

Con este diagrama se da conocer la arquitectura de la aplicación, donde el usuario a través de un dispositivo móvil accede a la información contactos (SQLite) y los videos (carpeta /res/raw).

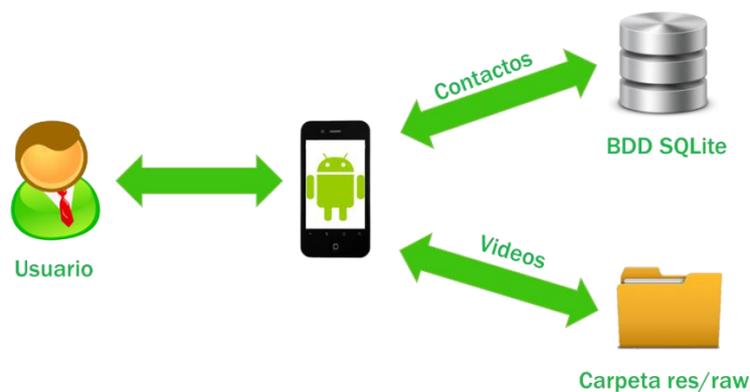


Figura 45: Diagrama arquitectónico

Fuente: propia

- Diagrama de navegación

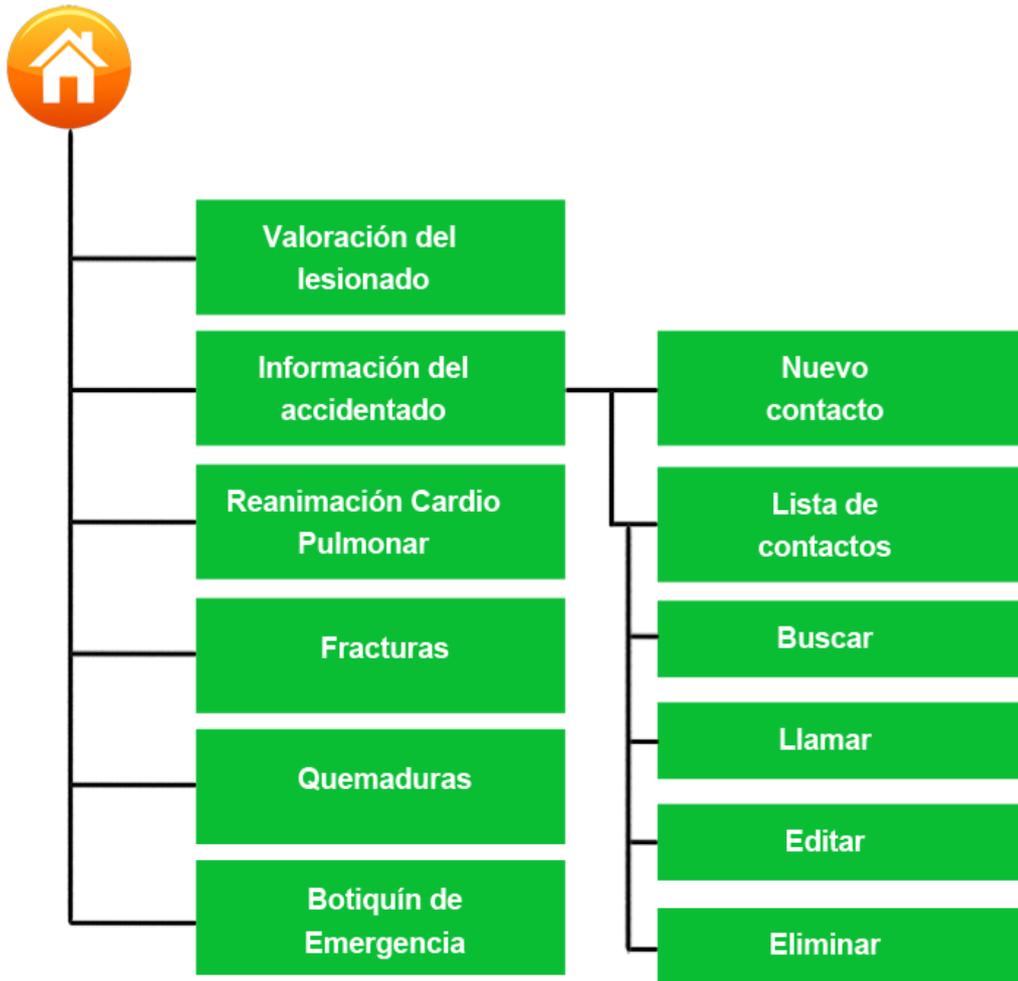


Figura 46: Diagrama de navegación

Fuente: propia

- Diseño de bocetos de las pantallas de la aplicación

Es necesario tener una idea del diseño de las interfaces gráficas, una interfaz amigable llama la atención del usuario.

Así como los arquitectos necesitan diseñar planos para empezar la obra de igual manera los programadores, desarrolladores tienen que crear esquemas, dibujos que facilitan el desarrollo del proyecto.

Para el diseño de los bocetos se utilizó el programa llamado WireframeSketcher Studio versión: 4.0.0 (versión de prueba), se lo puede obtener de la siguiente dirección: <http://wireframesketcher.com/>



Figura 47: Bocetos de las pantallas de la aplicación

Fuente: propia

3.2.2.3 IMPLEMENTACIÓN

Como primera actividad de este ciclo se dedicó a la implementación de las clases que conforman el desarrollo de la aplicación con sus métodos, las clases en Android son conocidas como Actividades, cada actividad creada tiene que estar en el AndroidManifest para mostrar su contenido.

Como siguiente actividad se implementaron las interfaces gráficas que forman parte de la aplicación, en la fase anterior se diseñaron los bocetos que ayudaron a tener una idea

más clara en relación al diseño, para la creación de las mismas Android ofrece la utilización del lenguaje XML o también por código Java.

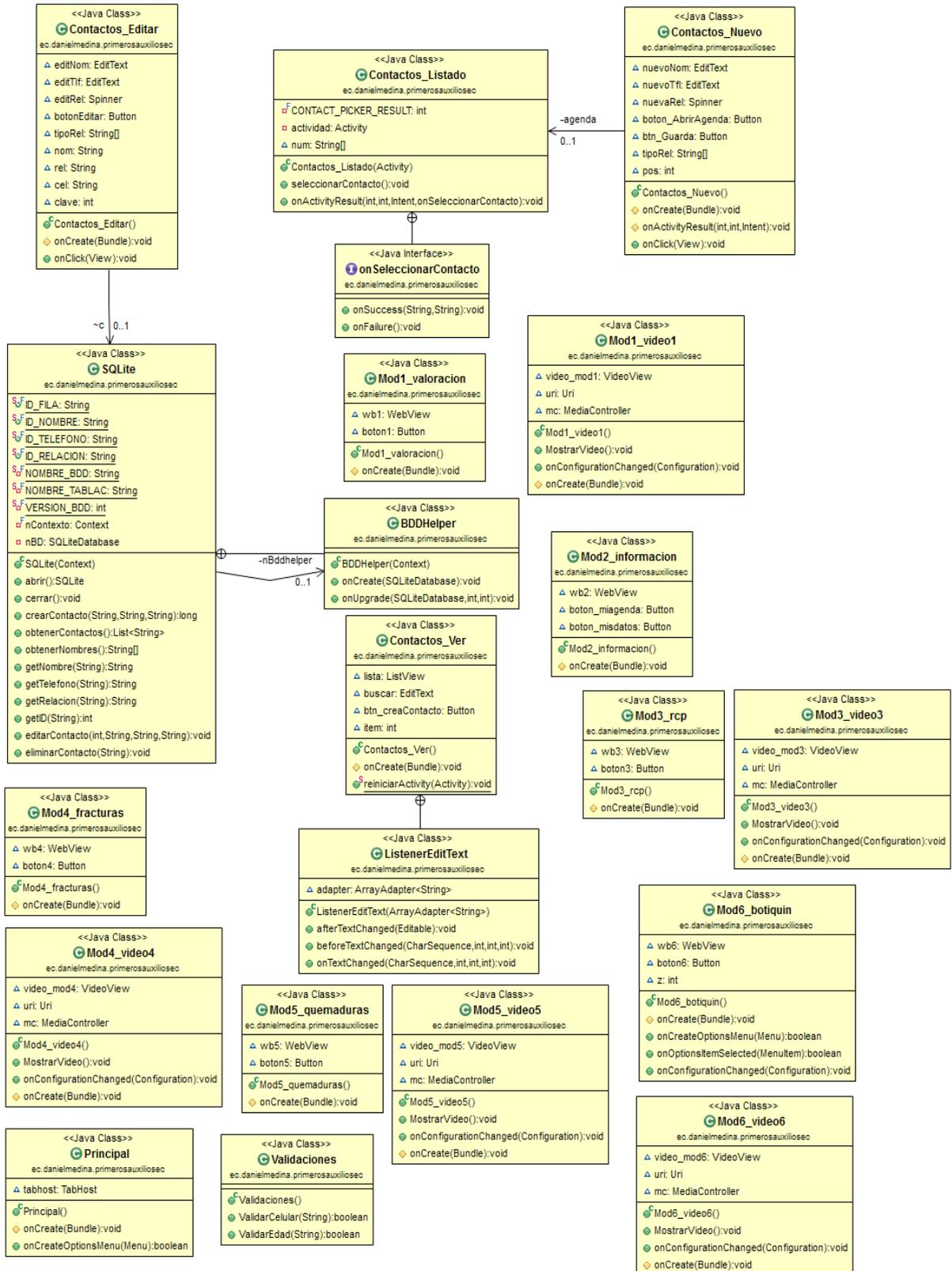


Figura 48: Diagrama de clases

Fuente: propia

• Interfaces gráficas

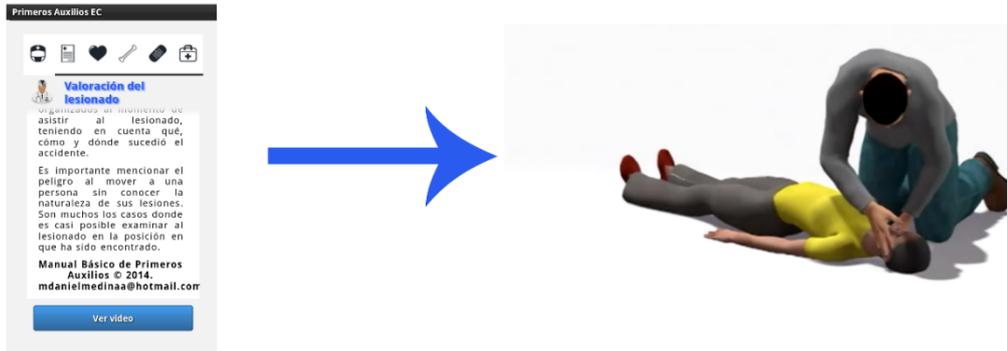


Figura 49: Interfaces valoración del lesionado

Fuente: propia



Figura 50: Interfaces de información del accidentado

Fuente: propia



Figura 51: Interfaces de reanimación cardio pulmonar

Fuente: propia



Figura 52: Interfaces de fracturas

Fuente: propia

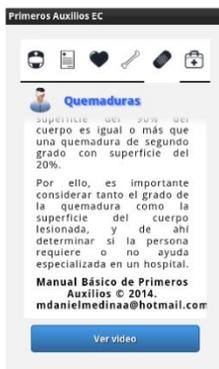


Figura 53: Interfaces de quemaduras

Fuente: propia

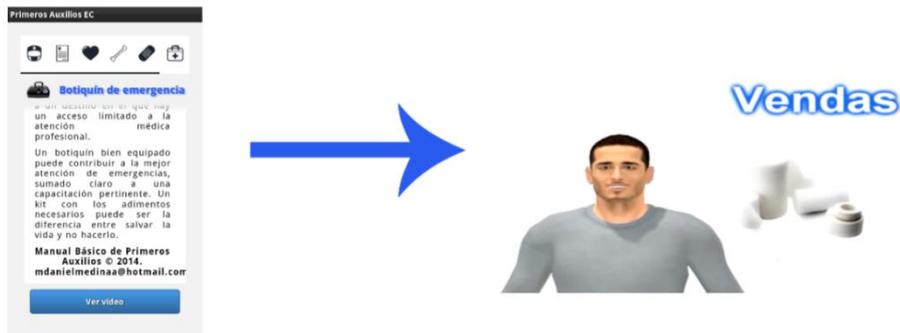


Figura 54: Interfaces de botiquín de emergencia

Fuente: propia

3.2.3 FASE APRENDER

3.2.3.1 PRUEBAS

Sin lugar a duda todo software tiene que ser sometido a un proceso de pruebas, la aplicación Primeros Auxilios EC está considerada dentro de la categoría de educación, salud y bienestar para todos los usuarios que poseen un dispositivo móvil con el sistema operativo Android versión 2.3 (Gingerbread).

La aplicación está considerada como un manual básico de primeros auxilios mediante animaciones, es parte de una investigación por parte del postulante que se adentra al mundo de la programación móvil.

• Introducción

Este documento tiene como objetivo indicar las pruebas a las cuales tiene que ser sometida la aplicación, cumpliendo así con los requerimientos establecidos en la fase de análisis.

Con el proceso de plan de pruebas se pretende que la aplicación en sí con la mayoría de los módulos sean probados una vez realizado la codificación pertinente y no esperar a su finalización.

El desarrollo de la aplicación se basó en la metodología ASD, lo que permite realizar pequeñas soluciones de una forma independiente. Esta metodología plantea una característica muy importante que es el feedback (retroalimentación) en la fase del aprendizaje que ayuda a comprender mejor el dominio del proyecto y satisfacer las necesidades del cliente permitiendo al programador y desarrollador corregir los errores a tiempo porque conforme avanza el proyecto el volumen de código aumenta.

En la tabla 18 se muestra los dispositivos en los cuales se realizaron las pruebas para diferentes versiones y tamaños de pantalla.

Nro.	Dispositivo	Versión	Tamaño
1	 <p data-bbox="427 600 860 633">Sony Ericsson Xperia Play R800i</p>	2.3.4	4.0 pulgadas 480x854 pixeles
2	 <p data-bbox="488 880 799 913">Samsung Galaxy Tab 3</p>	4.1	7.0 pulgadas 600x1024 pixeles
3	 <p data-bbox="504 1189 783 1223">LG Optimus L3 E400</p>	2.3.6	3.2 pulgadas 240x320 pixeles
4	 <p data-bbox="453 1464 831 1498">Samsung Galaxy Ace S5830</p>	2.3	3.5 pulgadas 320x480 pixeles
5	 <p data-bbox="469 1765 815 1798">Samsung T989 Galaxy S2</p>	2.3	4.52 pulgadas 480x800 pixeles

Tabla 18: Dispositivos móviles de pruebas

Fuente: propia

• **Plan de verificación de los requerimientos**

Con este plan se asegura el cumplimiento de los requerimientos planteados en el proyecto para el correcto funcionamiento, desde el punto de vista del programador y desarrollador.

Requerimiento	Programador	Responsables de verificación	Implementado	Observaciones
Reproducir los videos en toda la pantalla, la orientación debe ser horizontal.	Daniel Medina	Daniel Medina Director de Tesis	Si	Ninguna
La información al usuario debe ser clara y precisa con personajes 3D virtuales.	Daniel Medina	Daniel Medina Director de Tesis	Si	Se debe mejorar la iluminación y cambiar el color de fondo.
El formato de video deber ser compatible con cualquier dispositivo móvil android.	Daniel Medina	Daniel Medina Director de Tesis	Si	Se debe buscar un programa para convertir los videos o animaciones.
Crear una agenda de contactos.	Daniel Medina	Daniel Medina Director de Tesis	Si	Ninguna
Agregar, editar, buscar, eliminar, llamar un contacto.	Daniel Medina	Daniel Medina Director de Tesis	Si	Se debe poner como prioridad a la opción de llamar.
Utilizar los datos (nombre, teléfono) de la agenda del dispositivo para agregar un contacto.	Daniel Medina	Daniel Medina Director de Tesis	Si	Ninguna

Tabla 19: Plan de verificación de los requerimientos

Fuente: propia

Pruebas funcionales

Las pruebas funcionales sirven para encontrar diferencias entre el software desarrollado y explicación funcional para evaluar los casos de uso planteados anteriormente.

Actividad	Verificar la gestión de datos de contactos: crear, buscar, llamar, editar, eliminar.	Verificar la información que se proporciona en las animaciones
Usuario	Varios	Varios
Responsables	Daniel Medina	Daniel Medina
Entradas	Nombre, teléfono, relación	Ver video
Salidas	Contacto ingresado, datos actualizados, contacto eliminado, desea realizar la llamada.	Reproducción del video en toda la pantalla del dispositivo.
Observaciones	Para llamar, editar, eliminar primero se debe buscar el contacto en la lista.	Para reproducir un video se debe escoger un módulo.
Estado	Aprobado	Aprobado

Tabla 20: Pruebas funcionales

Fuente: propia

3.2.3.2 DOCUMENTACIÓN

La documentación realizada está detallada en los anexos: manual técnico y de usuario.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 ANÁLISIS DE COSTO BENEFICIO

Para realizar un análisis del costo de un proyecto, se debe considerar desde lo más pequeño (esfero, lápiz, borrador, entre otros) hasta lo más grande que se ha utilizado (laptop, computador de escritorio, entre otros), además cabe señalar que para el desarrollo de la aplicación se utilizaron herramientas de software libre con el propósito de que el costo no sea muy elevado.

Descripción	Presupuesto	Valor real
- Dispositivos móviles (smartphone y tablet)	660	660
- Laptop	1100	1100
- Computador de escritorio	700	700
- Aceleradora gráfica	54	54
- Discos duros (500GB)	180	180
TOTAL	2 694	2 694

Tabla 21: Costos de hardware
Fuente: propia

Descripción	Presupuesto	Valor real
- SDK de Android	0	660
- IDE Eclipse	0	1100
- Plugin ADT	0	700
- API's Android	0	0
- Poser Pro 2012	500	500
- Convertidores y editores de video	0	0
TOTAL	500	500

Tabla 22: Costos de software
Fuente: propia

Descripción	Presupuesto	Valor real
- Papel	25	25
- Esferos (\$0,30*2)	0,60	0,60
- Copias	5	5
- Impresiones de documentos	60	60
- Anillados	5	5
- Cuaderno	0,80	0,80
TOTAL	96,40	96,40

Tabla 23: Costos de suministros

Fuente: propia

Descripción	Presupuesto	Valor real
- Internet (\$25 * 12 meses)	300	300
- Transporte (\$10 cada semana por 6 meses)	240	240
- Imprevistos	150	150
- Capacitación adicional	200	200
- Subir la aplicación al Play Store	25	25
TOTAL	915	915

Tabla 24: Costos varios

Fuente: propia

Descripción	Valor real
- Costos de hardware	2 694
- Costos de software	500
- Costos de suministros	96,40
- Costos varios	915
TOTAL	4 205,40

Tabla 25: Costo total

Fuente: propia

Cabe mencionar que los costos detallados en las tablas anteriores, corrieron por cuenta propia del autor del proyecto, el IESS como entidad auspiciante no tuvo que invertir ningún valor para la implementación y desarrollo del mismo.

En cuanto a los beneficios están reflejados en el aprendizaje de los usuarios sobre las actuaciones de emergencia llamadas primeros auxilios a través de videos o animaciones, brindando información de lo que se debe hacer si existe algún accidente.

Al presentarse una situación de emergencia, el usuario puede actuar con la finalidad de mantener al accidentado fuera de peligro hasta disponer de personas profesionales, es decir que posean una mayor capacitación y conocimiento.

Otro de los beneficios es, que el usuario contará con una agenda de contactos a los cuales se los podrá localizar mediante una llamada, ya que en cualquier accidente la persona que brinda la ayuda podrá informar a un familiar sobre lo acontecido.

Referente al costo de la aplicación el usuario no tendrá que pagar ningún valor para su descarga e instalación, se la puede obtener gratuitamente del Play Store. Para su funcionamiento no necesita estar conectado a Internet, la aplicación es off-line.

4.2 IMPACTOS

4.2.1 IMPACTO SOCIAL

Las personas deben poseer un conocimiento básico sobre los primeros auxilios, ya que no se sabe en qué momento puede ocurrir algún accidente, le puede pasar a cualquier persona e incluso a un familiar, es por eso que la aplicación brinda una ayuda, transmitiendo información de actuaciones inmediatas en forma visual por si existe alguna situación de emergencia.

4.2.2 IMPACTO AMBIENTAL

Al ser una aplicación para dispositivos móviles no existe la necesidad de utilizar papeles para difundir la información a través de volantes, periódicos, revistas, entre otros; aportando así el cuidado del medio ambiente reduciendo la tala de árboles.

4.2.3 IMPACTO TECNOLÓGICO

Cada vez son más los dispositivos móviles y aplicaciones que salen al mercado, éstas cumplen con las necesidades que tienen los usuarios en el diario vivir, motivo por el cual se desarrolló este proyecto con tecnología Android y animaciones con personajes 3D virtuales.

4.3 CONCLUSIONES

Luego de terminar el proyecto de investigación, se concluye que:

- Los dispositivos móviles son fáciles de transportar debido a su pequeño tamaño, en estos dispositivos es posible instalar todo tipo de aplicaciones que satisfagan las necesidades del usuario ya sean gratuitas o de pago.
- Se consiguió crear la aplicación de Primeros Auxilios para teléfonos y tablets con tecnología Android, mostrando videos o animaciones 3D virtuales mediante el uso del software Poser Pro 2012.
- Se logró adquirir nuevos conocimientos sobre el uso del software Poser utilizado para crear las animaciones integradas en el dispositivo mostrando una visualización agradable al usuario con muy buena calidad.
- La elaboración de las animaciones, es un trabajo muy duro que requiere de mucho tiempo para llegar a mostrar el resultado final que cumpla con los requerimientos del cliente.
- La utilización de la metodología ASD permite crear un software de calidad a corto y mediano plazo, debido a que siempre se está en contacto con el cliente y el equipo de trabajo para corregir los errores existentes de forma rápida y poder continuar con el desarrollo sin tener problemas al momento de su finalización.
- Para la elaboración de este proyecto se ha logrado utilizar algunas herramientas gratuitas proporcionando la ventaja de tener licencias sin costo alguno.
- Programar aplicaciones para dispositivos que poseen el sistema operativo móvil Android por parte del autor ha sido una experiencia nueva y muy interesante, porque se adquirió nuevos conocimientos que servirán en la vida profesional. Es una gran ventaja realizar aplicaciones con software libre, es decir que las personas que deseen realizar algún proyecto basado con esta tecnología pueden tener todas las herramientas de forma gratuita.

4.4 RECOMENDACIONES

- Se debe tener muy en cuenta todas las características del dispositivo en especial la versión del sistema operativo y el tamaño, ya que las aplicaciones desarrolladas deben funcionar en cualquier dispositivo y adaptarse a cualquier tipo de pantalla sean pequeñas, medianas o grandes.

- Para incursionar en el mundo de Android hay que tener conocimientos básicos de programación en el lenguaje Java y POO²².
- Para cumplir con todos los requerimientos que se proponen por parte del usuario se recomienda mantener una comunicación constante con él para evitar problemas finales.
- Al momento de realizar una aplicación o un sistema para dispositivos móviles se recomienda utilizar una metodología ágil, ya que se obtiene un trabajo en conjunto por parte del desarrollador y del cliente.
- Se recomienda desarrollar una aplicación Android con una versión API baja con el propósito de que pueda funcionar en la mayoría de dispositivos.
- Por experiencia del autor se recomienda tener respaldos de todo lo que se refiere a documentación y el aplicativo sea en la nube (DropBox, SkyDrive) o en medios físicos (CD's, DVD's, discos duros externos, memorias flash), porque puede ocurrir algún imprevisto perdiendo información y tiempo para su culminación.
- Cuando se pretende ejecutar la aplicación para revisar su funcionamiento no se debe utilizar solo el emulador, es necesario tener dispositivos reales con diferentes características que posean el sistema operativo Android.
- Es muy aconsejable tener esas ganas de seguir aprendiendo cada día más, se tiene que ser investigativo ya que la tecnología avanza a pasos gigantescos. Si existen algunos problemas en el desarrollo se recomienda revisar la documentación oficial y en foros, en estos se pueden encontrar respuestas y manifestar inquietudes para solucionar los mismos.

²² POO. Programación Orientada de Objetos

BIBLIOGRAFÍA

Libros

Báez, M., Borrego, Á., Cordero, J., Cruz, L., Gonzáles, M., Hernández, F., . . . Zapata, Á. (2012). *Introducción a Android*. Madrid: UCM. Recuperado el 27 de Mayo de 2013, de <http://www.etnassoft.com/biblioteca/introduccion-a-android/>

DiMarzio, J. (2008). *Android A Programmer's Guide*. New York, Chicago, San Francisco: McGraw-Hill. doi:10.1036/0071599886

Lee, W.-M. (2011). *Android 4. Desarrollo de aplicaciones* (Española ed.). (B. Parra Pérez, Trad.) Madrid: Grupo Anaya S.A.

BIBLIOGRAFÍA

PUBLICACIÓN EN LINEA

[WEB1]

Haro, L. d. (8 de Diciembre de 2012). *Clasificación de dispositivos móviles*. Recuperado el 15 de Abril de 2013, de Prezi: <http://prezi.com/r0gwsopl3xmr/clasificacion-de-dispositivos-moviles/>

[WEB2]

Android domina en mercado mundial de celulares el segundo trimestre, s. a. (2012 de Agosto de 2012). *Sistemas operativos en teléfonos: Ventas y participación entre abril y junio de 2012*. Obtenido de emol: <http://www.emol.com/noticias/economia/2012/08/08/554717/android-domina-mercado-mundial-de-celulares-el-segundo-trimestre-superando-a-ios.html>

[WEB3]

Reyes, M. (6 de Marzo de 2013). *Los 5 mejores Sistemas operativos para celulares*. Obtenido de iPhoneando RD: <http://iphoneandord.com/los-5-mejores-sistemas-operativos-para-celulares/>

[WEB4]

Android Developers. (15 de Enero de 2013). Obtenido de <http://developer.android.com/about/index.html>

[WEB5]

Palou, N. (10 de Octubre de 2013). *La historia de Android resumida en un gráfico*. Recuperado el 4 de Noviembre de 2013, de Microsiervos: <http://www.microsiervos.com/archivo/gadgets/historia-android-resumida-grafico.html>

[WEB6]

Aranaz Tudela, J. (2009). *Desarrollo de Aplicaciones para Dispositivos Móviles sobre la plataforma Android de Google*. Ingeniero en Informática. Universidad Carlos III de Madrid Escuela Politécnica Superior.

[WEB7]

Universidad de Las Palmas de Gran Canaria. (5 de Abril de 2010). *Oficina del Software Libre - OSL*. Recuperado el 1 de Diciembre de 2011, de Documentación del curso de Desarrollo Android: <http://osl.ulpgc.es/cursos/android>

[WEB8]

Vico, Á. (17 de Febrero de 2011). *Arquitectura de Android*. Recuperado el 13 de Febrero de 2013, de La columna 80: <http://columna80.wordpress.com/2011/02/17/arquitectura-de-android/>

[WEB9]

Oliver, S. G. (2011). *Curso Programación Android v2*. Recuperado el 13 de Junio de 2012, de OpenLibra: <http://www.etnassoft.com/biblioteca/curso-programacion-android-v2/>

[WEB10]

Mebibyte. (1 de Octubre de 2013). Obtenido de Wikipedia: <http://es.wikipedia.org/wiki/Mebibyte>

[WEB11]

Universidad Politécnica de Valencia. (15 de Noviembre de 2011). *Android: Programación de aplicaciones para móviles*. Obtenido de <http://www.androidcurso.com/index.php/curso-android-basico/tutoriales-android-basico/34-unidad-3-actividades-e-intenciones/130-las-intenciones>

[WEB12]

Universidad Carlos III. (s.f.). *Software de Comunicaciones*. Recuperado el 21 de Junio de 2012, de <https://sites.google.com/site/swcuc3m/home/android/api/librerias-basicas-interfaces-usuario>

[WEB13]

Universidad Politécnica de Valencia. (2011). *Layouts*. Obtenido de <http://www.androidcurso.com/index.php/tutoriales-android/32-unidad-2-diseno-de-la-interfaz-de-usuario-vistas-y-layouts/114-layouts>

[WEB14]

SlideShare Inc. (13 de Enero de 2012). *Componentes de una aplicación*. Recuperado el 20 de Julio de 2013, de http://www.slideshare.net/android_unam/android-de-la-a-a-la-z-componentes-de-una-aplicacion-ulises-gonzalez

[WEB15]

Varios. (2012). *Curso de Programación en Android para Principiantes*. (R. P, Ed.) Recuperado el 28 de Mayo de 2012, de <http://www.etnassoft.com/biblioteca/curso-de-programacion-en-android-para-principiantes/>

[WEB16]

Vogel, L. (19 de Agosto de 2013). *Android base de datos y el contenido SQLite proveedor - tutorial*. Obtenido de <http://www.vogella.com/articles/AndroidSQLite/article.html>

[WEB17]

Smith Micro Software, Inc. (20 de Marzo de 2013). *Poser & Poser Pro User Manuals*. Obtenido de https://support.smithmicro.com/app/answers/detail/a_id/2281

ANDROID

{francho(lab)}. (2011). Android: usando contactscontract para sacar el listado de teléfonos de la agenda. [En línea] Obtenido de: <http://francho.org/2011/01/11/android-usando-contactscontract-para-sacar-el-listado-de-telefonos-de-la-ageda/> [Consultado: 12 de enero 2013]

Alcalde, A. (2013). *Optimizando la interfaz android - compound drawables*. [En línea] Obtenido de: <http://elbauldelprogramador.com/optimizando-la-interfaz-android-compound-drawables/> [Consultado: 02 de junio 2013]

Androidcurso.com. (2011). Uso de tabhost - android: programación aplicaciones para móviles. [En línea] Obtenido de: <http://www.androidcurso.com/index.php/recursos-didacticos/tutoriales-android/32-unidad-2-diseno-de-la-interfaz-de-usuario-vistas-y-layouts/213-uso-de-tabhost> [Consultado: 08 de septiembre 2013]

Androidcurso.com. (n.p.). Leer información de un contentprovider - android: programación aplicaciones para móviles. [En línea] Obtenido de: <http://www.androidcurso.com/index.php/curso-android-basico/tutoriales-android-basico/42-unidad-9-almacenamiento-de-datos/317-leer-informacion-de-un-contentprovider> [Consultado: 09 de abril 2013]

Androideity.com. (2011). Aplicar estilos y temas en android | androideity. [En línea] Obtenido de: <http://androideity.com/2011/09/14/aplicar-estilos-y-temas-en-android/> [Consultado: 12 de junio 2012]

Androideity.com. (2012). Soporte para diferentes dispositivos en android i | androideity. [En línea] Obtenido de: <http://androideity.com/2012/06/05/soporte-para-diferentes-dispositivos-en-android-i/> [Consultado: 24 de febrero 2013]

Androideity.com. (2012). Ui fluidas y la propiedad weight en android | androideity. [En línea] Obtenido de: <http://androideity.com/2012/06/01/ui-fluidas-y-la-propiedad-weight-en-android/> [Consultado: 05 de enero 2013]

Aprendeandroid.com. (2012). Aprende android.com – interface de usuario, los botones: button, imagebutton, togglebutton. [En línea] Obtenido de: <http://www.aprendeandroid.com/l4/interface2.htm> [Consultado: 19 de marzo 2013]

Caracol.com.co. (2013). Android lideró el mercado de los sistemas operativos de teléfonos inteligentes en 2012. [En línea] Obtenido de: <http://www.caracol.com.co/noticias/tecnologia/android-lidero-el-mercado-de-los-sistemas-operativos-de-telefonos-inteligentes-en-2012/20130128/nota/1832099.aspx> [Consultado: 29 de abril 2013]

Developer.android.com. (n.p.). Supported media formats | android developers. [En línea] Obtenido de: <http://developer.android.com/guide/appendix/media-formats.html> [Consultado: 23 de octubre 2013]

León, J. (2012). Pestañas en android usando eclipse | androideity. [En línea] Obtenido de: <http://androideity.com/2012/02/13/pestanas-en-android-usando-eclipse/> [Consultado: 21 de mayo 2013]

Nosoandroid.blogspot.com. (2013). Aprendiendo android: cambiar color del botón al pulsarlo. [En línea] Obtenido de: <http://nosoandroid.blogspot.com/2013/01/cambiar-color-al-pulsar-boton.html> [Consultado: 06 de julio 2013]

Programando-android.blogspot.com. (n.p.). *Programación y actualidad android: como dar estilo: color, sombras, tamaño a un texto textview.* [En línea] Obtenido de:

<http://programando-android.blogspot.com/p/como-dar-estilo-color-sombras-tamano-un.html> [Consultado: 03 de agosto 2013]

Sgoliver.net. (2011). Interfaz de usuario en android: pestañas (tabs) | sgoliver.net blog. [En línea] Obtenido de: <http://www.sgoliver.net/blog/?p=2112> [Consultado: 21 de septiembre 2013]

ASD

Arancibia, Y., Gutiérrez, M., Pincheira, G., & Venegas, F. (14 de Septiembre de 2007). *Ingeniería de Software - Informe de metodología*. Obtenido de <http://grupocrystal.files.wordpress.com/2007/10/informe-de-metodologia.pdf>

Huapaya Escobedo, J. L. (14 de Mayo de 2013). *Metodología ASD*. Obtenido de Scribd: <http://es.scribd.com/doc/141283216/Asd#download>

Lasso, D. L., Medina, A. P., & Vidal, I. A. (13 de Junio de 2010). *ASD (Adaptive Software Development)*. Obtenido de Slideshare: <http://www.slideshare.net/urumisama/metodologia-agil-asd>

Ochoa Gonzáles, N. (24 de Junio de 2013). *Metodología ASD*. Obtenido de Prezi: <http://prezi.com/2iws5li2ve5j/untitled-prezi/>

POSER PRO

Cejas Pulido, J. (2010). Bryce juani cejas pulido. [En línea] Obtenido de: <http://jrcejaspulido.eresmas.net/bryce.htm> [Consultado: 15 de agosto 2013]

Conde, J. (2011). Curso de poser práctico. [Video en línea] Obtenido de: <http://www.youtube.com/watch?v=X1BtGI3ECP0&list=PL73264793384DA15E> [Fecha de consulta: 25 de enero 2013]

Mérida, A. (2013). Librerías de poser 2012 gratis, instala y descarga. [Video en línea] Disponible en: <http://www.youtube.com/watch?v=iDtMj0lvuvM> [Fecha de consulta: 12 de septiembre 2013]

Neoteo. (2012). Poser pro: primeros pasos y creación de ilustración - neoteo. [En línea] Obtenido de: <http://www.neoteo.com/poser-pro-primeros-pasos-y-creacion-de-ilustracion/> [Consultado: 20 de febrero 2013]

Poser.smithmicro.com. (2013). Poser tutorials - learn the basics of 3d animations. [En línea] Obtenido de: <http://poser.smithmicro.com/tutorials.html> [Consultado: 14 de octubre 2013]

Tutorial-enlace.net. (n.p.). Los mejores tutoriales poser gratis. [En línea] Obtenido de: <http://www.tutorial-enlace.net/top-tutorial-Poser.html> [Consultado: 17 de junio 2013]

Wikipedia. (2013). Gráficos 3d por computadora. [En línea] Obtenido de: http://es.wikipedia.org/wiki/Gr%C3%A1ficos_3D_por_computadora [Consultado: 19 de febrero 2013]

PRIMEROS AUXILIOS

Canalibermutuamur Ibermutuamur. (2013). *Ibermutuamur primeros auxilios fracturas*. [Video en línea] Disponible en: <http://www.youtube.com/watch?v=3NFdUF5I9H0> [Fecha de consulta: 5 de julio 2013]

CharHadas Tv. (2012). *Qué debe tener un botiquín de emergencias para casa o viaje*. [Video en línea] Disponible en: <http://www.youtube.com/watch?v=pd77agqoLQg> [Fecha de consulta: 15 de junio 2013]

Di Benedictis, J. (2013). *Cómo tratar quemaduras: primeros auxilios*. [Video en línea] Disponible en: http://www.youtube.com/watch?v=s3zy_fDEOKE [Fecha de consulta: 26 de agosto 2013]

FerreHogar. (2012). *Botiquín de primeros auxilios "intermedio" para viajes más largos y grandes grupos de personas*. [Video en línea] Disponible en: <http://www.youtube.com/watch?v=iRiZ2GmH0iE> [Fecha de consulta: 4 de Mayo 2013].

Fortes Gómez, S. (13 de Septiembre de 2011). *Guía de atención a emergencias sanitarias en los centros educativos*. Recuperado el 25 de Octubre de 2012, de http://www.gobiernodecanarias.org/educacion/5/DGOIE/PublicaCE/docsup/Guia_emergencias_sanitarias.pdf

Martínez, N. (2011). *Maniobras rcp mayores de 8 años y adultos (detalle)*. [Video en línea] Disponible en: <http://www.youtube.com/watch?v=sE5jkKSHVy8> [Consultado: 18 de febrero 2013]

Martínez, N. (2011). *Primeros auxilios. Valoración ABC*. [Video en línea] Disponible en: <http://www.youtube.com/watch?v=RJeTvrAhsy0> [Fecha de consulta: 23 de enero 2013]

Martínez, N. (2011). *Quemaduras. Primeros auxilios*. [Video en línea] Disponible en: <http://www.youtube.com/watch?v=FZOWFi9N3DI> [Consultado: 25 de febrero 2013]

Mundo Asistencial. (2011). *Primeros auxilios: quemaduras*. [Video en línea] Disponible en: <http://www.youtube.com/watch?v=mB1LXIITdWQ> [Fecha de consulta: 25 de Marzo 2013]

Últimas Noticias Quito. (2012). Primeros auxilios: fractura. [Video en línea] Disponible en: <http://www.youtube.com/watch?v=ujlLVBD5nuk> [Fecha de consulta: 23 de enero 2013]

SQLite

Developer.android.com. (n.p.). Android.database.sqlite | android developers. [En línea] Obtenido de: <http://developer.android.com/reference/android/database/sqlite/package-summary.html> [Consultado: 13 de julio 2013]

Proyecto Simio. (2013). *Programación android, bases de datos ii*. [En línea] Obtenido de: <http://proyectosimio.wordpress.com/2013/09/22/programacion-android-bases-de-datos-ii/> [Consultado: 02 de diciembre 2013]

Stackoverflow.com. (n.p.). Android sqlite example - stack overflow. [En línea] Obtenido de: <http://stackoverflow.com/questions/12015731/android-sqlite-example> [Consultado: 25 de mayo 2013]

Techotopia.com. (n.p.). An android sqlite database tutorial - techotopia. [En línea] Obtenido de: http://www.techotopia.com/index.php/An_Android_SQLite_Database_Tutorial [Consultado: 05 de agosto 2013]

Vogel, L. (2013). Android sqlite database and content provider - tutorial. [En línea] Obtenido de: <http://www.vogella.com/tutorials/AndroidSQLite/article.html> [Consultado: 15 de noviembre 2013]

ANEXOS

ANEXO A MANUAL TÉCNICO

• Instalación de Poser Pro 2012

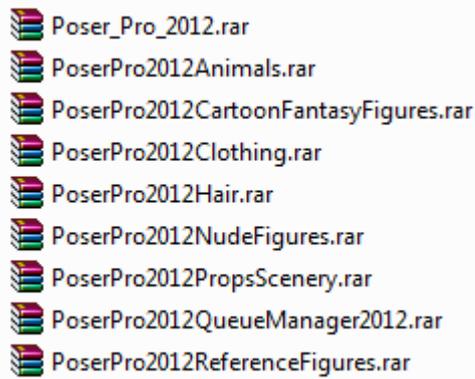
Poser Pro 2012, es un software para desarrollar animaciones 3D virtuales que cuenta con una librería de contenido.

• Requisitos del sistema (Windows)

Los requisitos para la instalación son los siguientes:

- Windows XP, Vista o 7 (64 bits del sistema operativo para la instalación de 64 bits)
- 1.3 GHz Pentium 4 o superior, Athlon 64 o superior (1.65 GHz o más rápido)
- 1 GB de memoria RAM del sistema (se recomienda 2 GB o más)
- OpenGL activado tarjeta gráfica o chipset recomendado (reciente NVIDIA GeForce y ATI Radeon necesarios para funciones de vista previa en tiempo real avanzadas)
- Pantalla en color de 24 bits, 1024 x 768 resolución mínima
- 3 GB de espacio libre en el disco duro (5 GB recomendado)
- Unidad de DVD-ROM
- Conexión a Internet
- Internet Explorer 7 o superior
- Adobe Flash Player 9 o posterior (Flash Player 11 necesario para la biblioteca introducida de 64 bits)
- Adobe AIR

La siguiente pantalla indica todos los archivos necesarios para realizar la instalación del software Poser Pro 2012:



• Pasos para la instalación

Para comenzar con la instalación se debe seguir los siguientes pasos:

1. Se tiene que dar doble clic en el instalador de Poser_Pro_2012 para iniciar con la instalación.



2. Aparecerá la pantalla de bienvenida, para lo cual se da clic en Next.

3. Luego aparece el acuerdo Poser licencia de usuario final, con la cual se da un clic en aceptar el acuerdo (I accept the agreement) y un clic en Next.

4. Aparece la pantalla de información, este archivo contiene información que puede afectar la forma de instalar y/o utilizar Poser. Cuando se haya terminado de leer el archivo Léame, se da clic en Next para continuar con la instalación.

5. Aparece la pantalla de selección de ubicación de destino, aquí se puede seleccionar la ubicación donde se quiere instalar Poser Pro dando clic en Browse, o se puede dejar por defecto y se da clic en Next.

6. Si el equipo y el sistema operativo son de 64 bits, el programa de instalación para dicha versión se instalará de forma predeterminada. También se puede instalar opcionalmente y adicionalmente la versión para equipos y el sistema operativo de 32 bits.

7. Aparece la pantalla para seleccionar los componentes. Existen las siguientes opciones en el menú desplegable:

- **Full Installation (Instalación completa):** la mayoría de usuarios eligen ésta opción de instalación, ya que ofrece una instalación completa incluyendo todo el contenido y el gestor de colas (Queue Manager).

- **Compact Installation (Instalación compacta):** ésta opción instala archivos de contenido Core Poser Pro 2012, el administrador de colas de red, archivos de soporte técnico de la aplicación y contenido adicional si se seleccionan las opciones aplicables. Ésta opción es útil si el espacio en disco es limitado.

- **Basic Content (Contenido Básico):** esta opción instala figuras vestidas, accesorios, paisaje, etc.

- **Custom Installation (Instalación Personalizada):** esta instalación permite seleccionar los elementos necesarios que se desea instalar.

- **Queue Manager Only:** instala solamente el Queue Manager Poser (gestor de colas).

8. Luego dar clic en Next para pasar a la pantalla de selección de ubicación del contenido. Ahí seleccionar la ubicación para el directorio Runtime donde se encuentra el contenido de Poser Pro. Se tiene las siguientes opciones:

- **Shared Documents directory:** esta es la opción de instalación predeterminada.

- **My Documents directory:** los archivos serán solo accesibles para el usuario que las instaló.

- **Poser directory:** la biblioteca de contenido reside junto a los archivos del programa Poser Pro.

- **Other location:** esta opción es conveniente si se desea instalar el contenido en un disco duro diferente o nombre de carpeta.

9. Luego aparece la pantalla con tareas adicionales, aquí permite eliminar, realizar copias de seguridad o reutilizar archivos de preferencia existentes que almacenan información como tiempos de ejecución de la biblioteca, diseños de interfaz y otras preferencias. También se puede activar las casillas de verificación para crear iconos para Poser (64 y 32 bits) y Queue Manager, luego dar clic en Next.

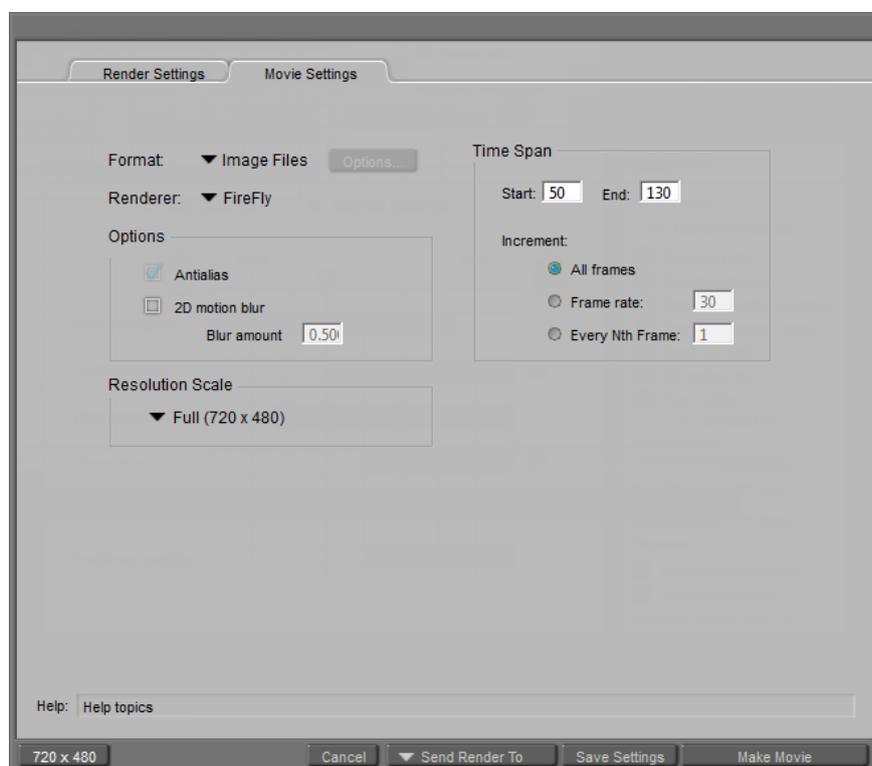
10. En la siguiente pantalla se revisa todas las opciones, se da clic en Install para instalar Queue Manager y Poser Pro. Cuando haya terminado la instalación se da clic en Finish para finalizar la instalación.

Para la instalación del contenido de la biblioteca como: Animals, Cartoon Fantasy Figures, Clothing, Hair, Nudes Figures, Props Scenary y Reference Figures, solo se debe

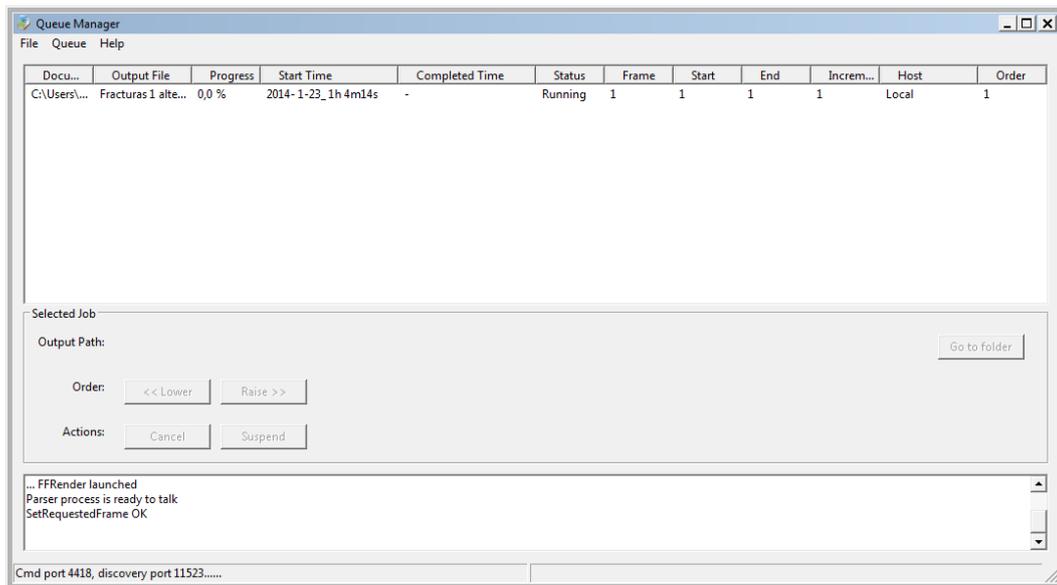
ejecutar y seguir los pasos de instalación básicos Next y seleccionando la ubicación del directorio Runtime escogiendo la opción Poser directory.

• Renderizado de animaciones

Una vez que se ha terminado de realizar la animación, se debe renderizar todos los fotogramas. Poser ofrece 3 formas de hacerlo (imágenes, película .avi y archivo flash). Para que la animación se visualice con alta calidad se debe renderizar mediante imágenes, pero este proceso es más largo, porque se crea una imagen por cada fotograma. Para acceder a Render Settings se debe presionar Ctrl+Y.



El gestor de cola Queue Manager es el que realiza este proceso para crear las imágenes por cada fotograma. Para ingresar se debe ir al menú Render → Render In Queue.



Con esto se tendrá toda la animación pero mediante imágenes. Para realizar la secuencia de imágenes se puede utilizar cualquier programa de edición de video como: Sony Vegas o Adobe Premiere.



- **Transformar los videos a formato Android**

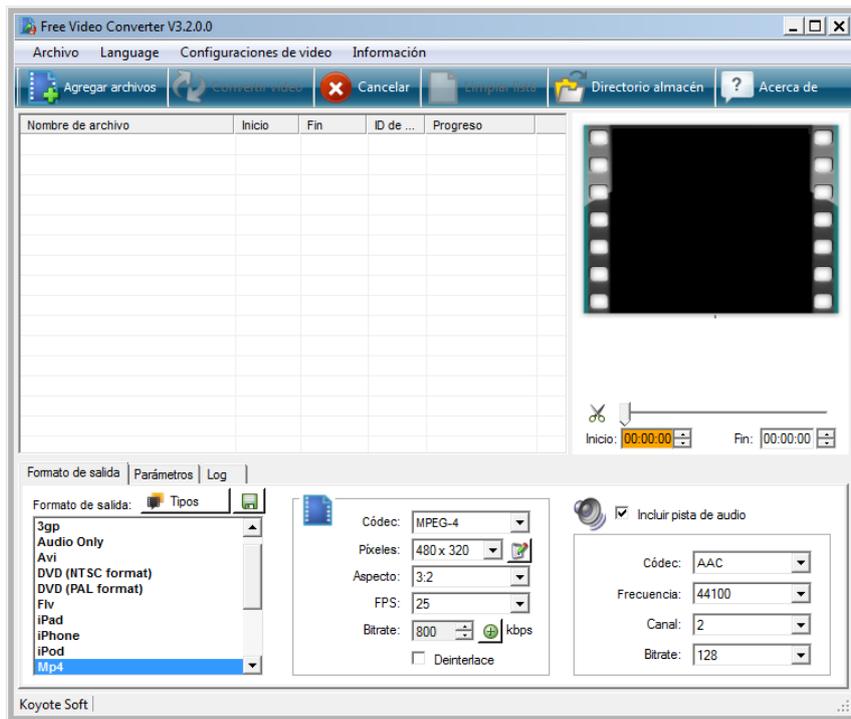
Para que funcionen los videos en la mayoría de dispositivos Android se puede utilizar el programa Free Video Converter, se lo puede obtener de: <http://www.koyotesoft.com/>

Free Video Converter
Convert videos to various formats
Save movies and parts of your videos or
extract music only

[Free Download](#)

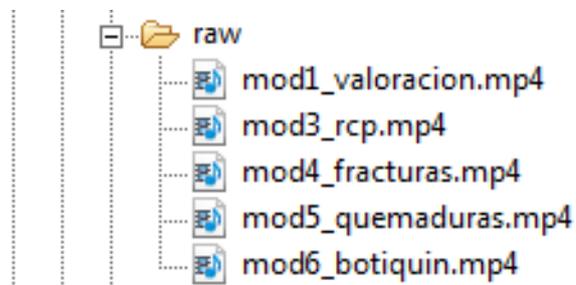


La configuración para transformar el video se muestra en la siguiente pantalla:



- **Insertar los videos en el proyecto Android**

Se debe crear la carpeta con el nombre raw dentro del directorio /res.



A continuación se muestra la parte del código fuente (.java y .xml) para visualizar los

```
public void MostrarVideo() {
    video_mod1 = (VideoView) findViewById(R.id.video_mod1);
    uri = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.mod1_valoracion);
    mc = new MediaController(this);
    video_mod1.setMediaController(mc);
    video_mod1.setVideoURI(uri);
    video_mod1.start();
    video_mod1.requestFocus();
}
```

videos.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#ffffff" >

    <VideoView
        android:id="@+id/video_mod1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_centerInParent="true" />

</RelativeLayout>
```

Para que los videos se visualicen en toda la pantalla de cualquier dispositivo, en el archivo AndroidManifest.xml, en la actividad del video se tiene que realizar unas configuraciones como ocultar el teclado y que se visualicen en forma horizontal.

```
<activity
    android:name="ec.danielmedina.primerosauxiliosec.Mod1_video1"
    android:configChanges="keyboard|keyboardHidden|orientation"
    android:screenOrientation="landscape"
    android:theme="@android:style/Theme.NoTitleBar.Fullscreen" >
</activity>
```

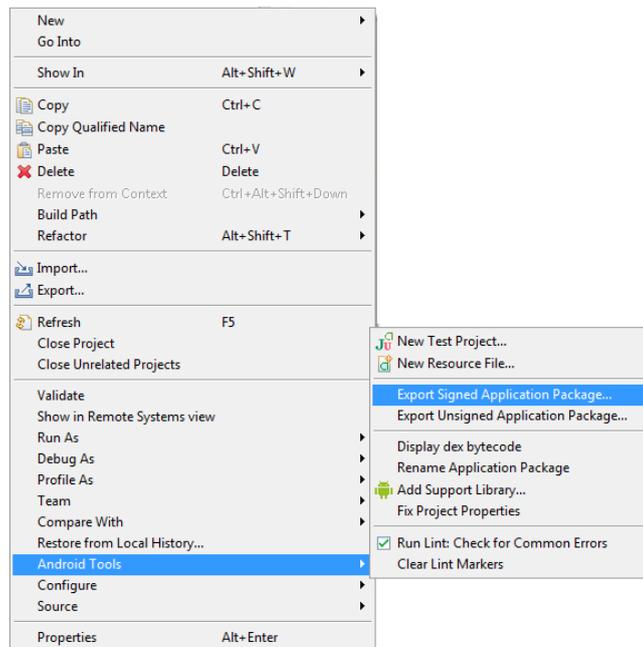
• Firmar la aplicación Android con el IDE Eclipse

Una vez que se ha finalizado con el desarrollo de la aplicación, uno de los requisitos es firmarla para poder publicarla en el Play Store.

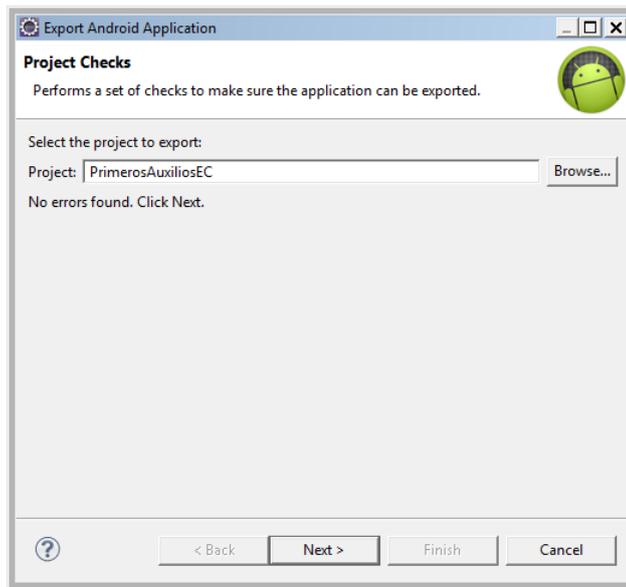
Cualquier aplicación debe ser firmada, esto se hace como medida de seguridad y requisito de garantía, evitando problemas para su respectiva distribución e instalación. La finalidad de firmar la aplicación es que el desarrollador sea el único que pueda modificar y actualizar la aplicación.

Pasos para firmar una aplicación

1. Se debe seleccionar el proyecto android a firmar, dar clic derecho sobre él y escoger la opción Android Tools → Export Signed Application Package...



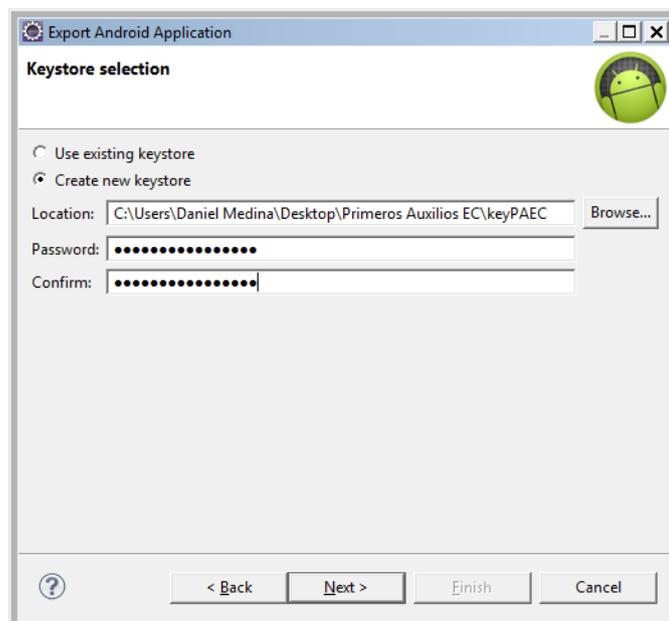
2. Aparece la siguiente pantalla, se da clic en Next



3. Es fundamental tener un keystore para firmar una aplicación, este es un almacén de claves donde se encuentran todos los certificados validados que se pedirán. Si la firma que se va a realizar es por primera vez, se debe seleccionar la opción Create new keystore y llenar los datos en los campos.

Location: se debe seleccionar el directorio donde se quiere guardar la keystore.

Password: se debe introducir una contraseña que contenga 6 dígitos o más para crearla.



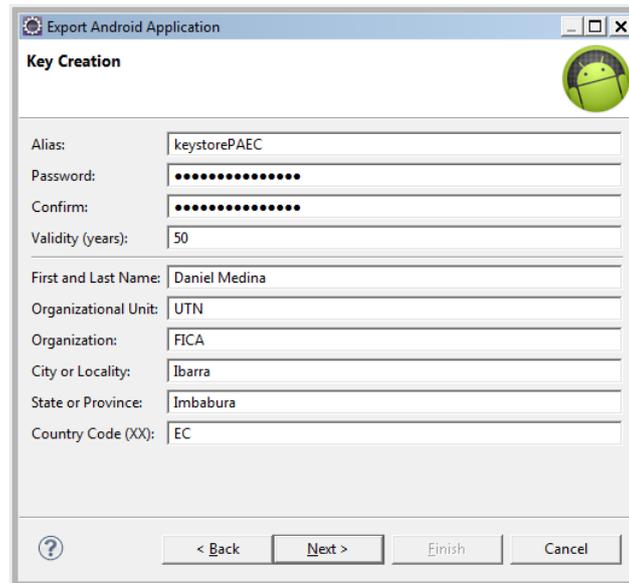
4. Se debe llenar los datos que presenta la siguiente pantalla.

Alias: puede ser el mismo nombre del proyecto pero abreviado.

Password: se asigna una contraseña.

Validity (years): se define la validación de la keystore en años.

Los campos que están a continuación se refieren a la información personal y de la organización o empresa.

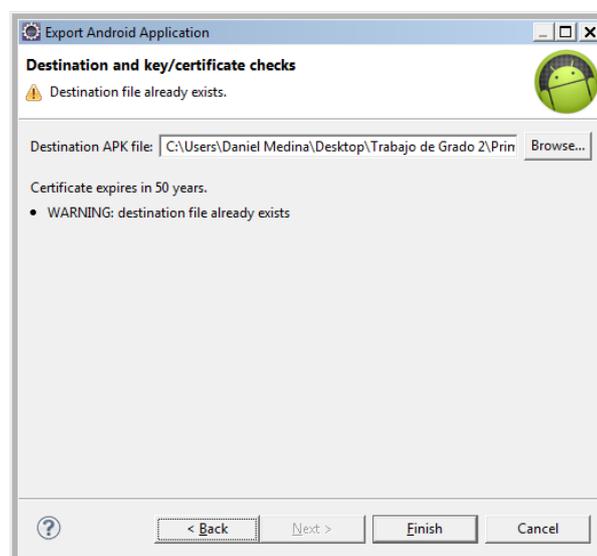


The screenshot shows the 'Export Android Application' dialog box, specifically the 'Key Creation' step. The dialog has a title bar with the text 'Export Android Application' and standard window controls. Below the title bar is the 'Key Creation' section, which includes an Android logo icon. The form contains the following fields and values:

- Alias: keystorePAEC
- Password: [Redacted]
- Confirm: [Redacted]
- Validity (years): 50
- First and Last Name: Daniel Medina
- Organizational Unit: UTN
- Organization: FICA
- City or Locality: Ibarra
- State or Province: Imbabura
- Country Code (XX): EC

At the bottom of the dialog, there are four buttons: a help icon (?), '< Back', 'Next >', 'Finish', and 'Cancel'.

5. En la siguiente pantalla, se escoge el directorio donde se quiere guardar el archivo .apk firmado, también indica el tiempo de expiración del certificado. Para terminar se da clic en Finish y se tendrá la aplicación lista para subir al Play Store.



The screenshot shows the 'Export Android Application' dialog box, specifically the 'Destination and key/certificate checks' step. The dialog has a title bar with the text 'Export Android Application' and standard window controls. Below the title bar is the 'Destination and key/certificate checks' section, which includes an Android logo icon and a warning icon. The form contains the following information:

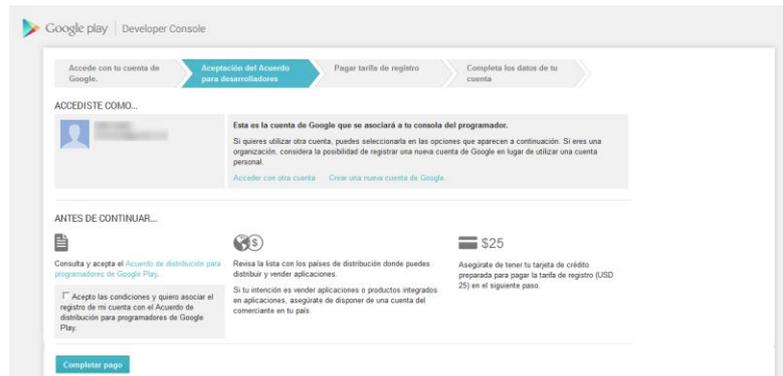
- Destination APK file: C:\Users\Daniel Medina\Desktop\Trabajo de Grado 2\Prin [Browse...]
- Certificate expires in 50 years.
- WARNING: destination file already exists

At the bottom of the dialog, there are four buttons: a help icon (?), '< Back', 'Next >', 'Finish', and 'Cancel'.

NOTA: Si se desea firmar otras aplicaciones basta seguir los pasos 1 y 2, pero ya no existe la necesidad de crear una nueva keystore, solo se debe utilizar la que ya está creada, por eso es muy importante guardar la contraseña.

Publicar la aplicación en el Google Play

Se debe ingresar con la cuenta de Google (Gmail) a la siguiente dirección:
<https://play.google.com/apps/publish/signup/>

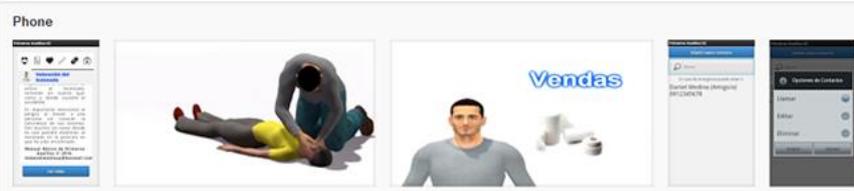


Para publicar la aplicación en el Google Play, se necesita pagar una única cuota de registro de por vida, el valor es de \$25, éste se lo realiza mediante Google Checkout.

A screenshot of the Google Wallet setup form. The page is titled "Google Wallet" and shows a browser address bar with "@gmail.com". The main heading is "Set up Google Wallet". Below this is a section "NAME AND HOME LOCATION" with a dropdown menu for "Ecuador (EC)". There are input fields for "Name", "Street address", "Postal code", and "City". Below this is a section "PAYMENT METHOD" with a heading "Credit or debit card". There is an input field for "Card number" with logos for VISA, MasterCard, AMEX, and DISCOVER. There are input fields for "Expiration date" (MM / YY) and "Security code" (CVC). There is a question mark icon next to the Security code field.

El desarrollador toma la decisión si la aplicación que se va a publicar es de pago o gratuita, pero se debe tener muy en cuenta la filosofía de Android, donde la gran mayoría de aplicaciones son gratuitas, por lo tanto se debe crear aplicaciones de pago si en realidad valen la pena porque son genuinas.

Se debe ingresar toda la información de la aplicación como: capturas de pantalla, el icono, el título, una descripción, texto promocional, entre otros. Y como último paso se debe subir la aplicación firmada para su distribución, el tiempo estimado para ver la aplicación es de 24 horas, igual si se realiza algún cambio o se tiene una nueva actualización.

Title * Spanish (Latin America) – es-419	Ecuador Primeros Auxilios 25 of 30 characters	
Description * Spanish (Latin America) – es-419	<p>RECUERDE: Los primeros auxilios no son tratamientos médicos. Son actuaciones de emergencia para reducir los efectos de las lesiones y estabilizar el estado del accidentado.</p> <p>Esta aplicación es un Manual Básico de Primeros Auxilios donde se puede observar animaciones indicando lo que se tiene que hacer en caso de una emergencia, permite crear contactos ya sea directamente o añadiéndolos de la agenda del teléfono.</p> <p>La aplicación se está desarrollando para un trabajo de grado, se pueden observar las animaciones de valoración del accidentado, reanimación cardio pulmonar, fracturas, quemaduras y botiquín de emergencia, en la información del accidentado se da la opción de crear una agenda de contactos que pueden ser amigos, familiares, primos entre otros a los cuales se los puede llamar en caso de alguna emergencia.</p>	
Phone		
<p>+ Add screenshot Drop image here.</p>		
High-res icon * Default – Spanish (Latin America) – es-419 512 x 512 32-bit PNG (with alpha)	Feature Graphic Default – Spanish (Latin America) – es-419 1024 w x 500 h JPG or 24-bit PNG (no alpha)	Promo Graphic Default – Spanish (Latin America) – es-419 180 w x 120 h JPG or 24-bit PNG (no alpha)
		

ANEXO B MANUAL DE USUARIO

Manual Básico de Primeros Auxilios

Introducción

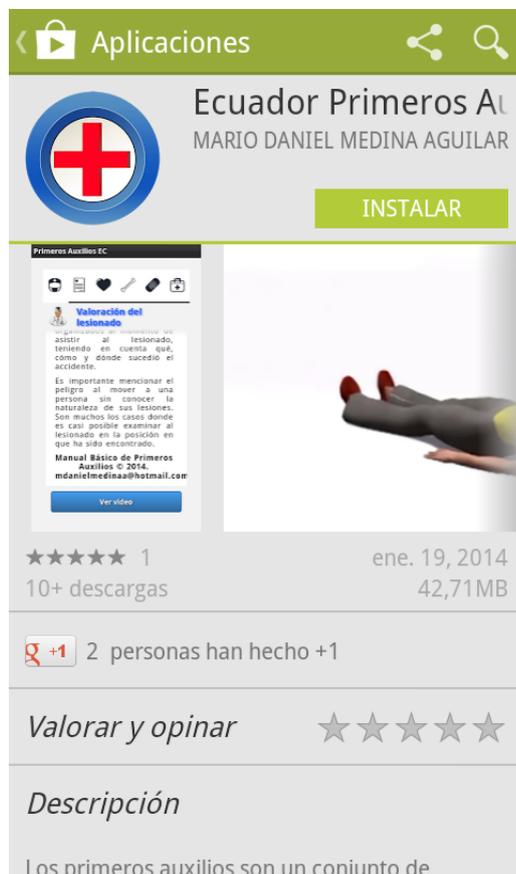
La aplicación Ecuador Primeros Auxilios brinda información sobre los primeros auxilios mediante la visualización de videos o animaciones, permite la creación de una agenda de contactos donde se añade el nombre, el número de celular y el tipo de relación (padre, madre, hermano/a y otros).

Usuarios

La aplicación está a disposición de los usuarios que poseen un dispositivo móvil con tecnología Android a partir de la versión 2.3 y que cuenten con la tarjeta SIM.

Ingreso a la aplicación

1. Para poder descargar e instalar la aplicación se debe ingresar al Play Store y buscar con el nombre de: Ecuador Primeros Auxilios.



2. Una vez instalada la aplicación se debe presionar en el icono:



**Primeros
Auxilios EC**

3. Al iniciar la aplicación se muestra la siguiente pantalla:



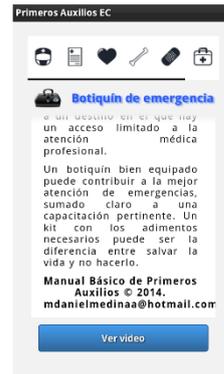
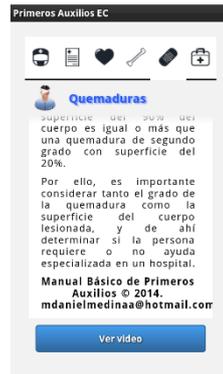
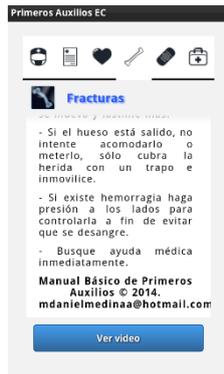
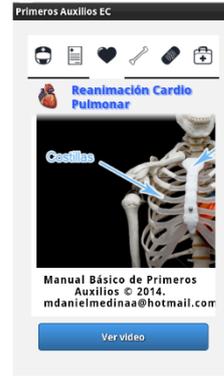
Ingreso a los módulos

Para el ingreso a los módulos simplemente se debe presionar el icono al que corresponda. La aplicación cuenta con los siguientes módulos:

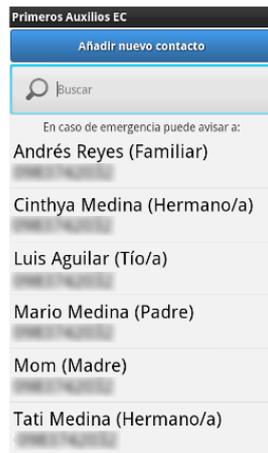
Icono	Módulo	Descripción
	Valoración del lesionado	Indica tres cosas fundamentales: el estado de alerta o conciencia la buena respiración y los signos de circulación.
	Información del lesionado	Permite la creación de una agenda de contactos con la finalidad de llamar a un familiar para comunicar lo acontecido.
	Reanimación Cardio Pulmonar	Se puede observar lo que se debe hacer si existe un paro cardíaco o respiratorio.
	Fracturas	Presenta ciertas indicaciones que se deben ejecutar por si existe una fractura sea abierta o cerrada.
	Quemaduras	Presenta una serie de recomendaciones que se deben realizar ante quemaduras de primer, segundo y tercer grado.
	Botiquín de emergencia	Se puede visualizar los elementos necesarios que debe tener un botiquín de viaje y en el hogar

Cada módulo presenta información tanto teórica como visual (videos o animaciones), a excepción del módulo información del accidentado.

El usuario para poder visualizar los videos debe desplazarse hasta la parte final y presionar el botón ver video. Para crear los contactos debe ir al módulo de información, de igual manera desplazarse al final y presionar el botón mis contactos.



Mis contactos: muestra el listado ordenado en forma ascendente de los contactos que han sido añadidos por parte del usuario.



Nota: Si se desea buscar un determinado contacto, simplemente se debe escribir en la caja de texto el nombre de la persona y se irá mostrando los resultados que correspondan.

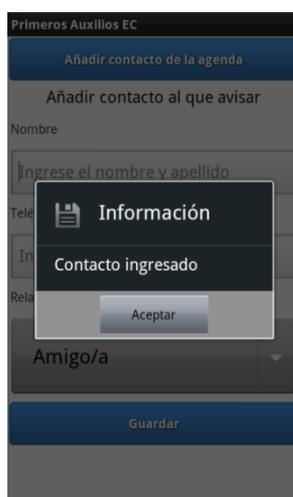
Añadir nuevo contacto: permite insertar los datos del contacto que se desea guardar en la agenda de la aplicación al cual se le puede avisar, estos son: nombre, celular y tipo de relación.



Nota: Si al guardar el contacto sin ninguna información, no se guardará nada y se indica que debe llenar la caja de texto con el nombre, también si el número de celular no es válido se notificará que el número no es el correcto.

Si se desea utilizar los datos del contacto del dispositivo simplemente se debe presionar el botón añadir contacto de la agenda, si existen varios números tiene la opción de escoger uno de ellos.

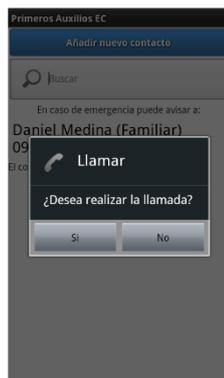
Notificación de contacto ingresado: una vez llenado todos los datos del contacto, se debe presionar el botón guardar y se notificará que el contacto ha sido ingresado.



Opciones de contactos: en listado de contactos al presionar un contacto se muestran las opciones de: llamar, editar y eliminar.

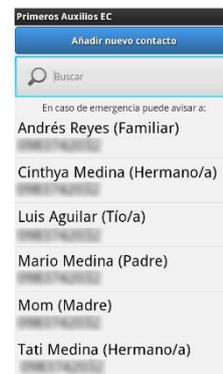


- **Llamar:** permite realizar una llamada al contacto seleccionado. Si no se desea realizar la llamada simplemente se debe presionar el botón No, con esto la llamada se cancelará.



Nota: Se debe tener saldo para hacer la llamada a cualquier contacto.

- **Editar:** permite actualizar cada uno de los datos del contacto.



- **Eliminar:** permite eliminar un contacto de la agenda de la aplicación.

