

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE CUENCA

CARRERA DE INGENIERÍA DE SISTEMAS

Tema:

“Diseño e Implementación de una Aplicación para Dispositivos Android en el Marco del Proyecto Pequeñas y Pequeños Científicos de la Universidad Politécnica Salesiana”

Tesis previa a la obtención del título de Ingeniero de Sistemas

Autor:

Diego Fernando Quisi Peralta

Director:

Ing. Vladimir Robles Bykbaev

Cuenca, Diciembre 2012

Ing. Vladimir Robles Bykbaev.

CERTIFICO

Haber dirigido y revisado prolijamente la tesis titulada “Diseño e Implementación de una Aplicación para Dispositivos Android en el Marco del Proyecto Pequeñas y Pequeños Científicos de la Universidad Politécnica Salesiana”, realizado por el estudiante Diego Fernando Quisi Peralta, y por cumplir los requisitos autorizo su presentación.

Cuenca, Diciembre de 2012

V.R.:

Ing. Vladimir Robles Bykbaev.
Director de Tesis

DECLARATORIA DE RESPONSABILIDAD

Yo Diego Fernando Quisi Peralta portador de la cedula de ciudadanía 010461646-1, estudiante de la carrera de Ingeniería de Sistemas, certifico que los conceptos desarrollados, análisis realizados, así como el desarrollo e implementación son de exclusiva responsabilidad del autor. Los textos de otros autores llevan su correspondiente cita bibliográfica y autorizo a la Universidad Politécnica Salesiana el uso de la misma para fines académicos.



Diego Fernando Quisi Peralta

DEDICATORIAS

Dedico esta tesis a Dios, por darme todas las bendiciones, salud y estar en mi vida siempre, a mi Madre Luisa Peralta de quien siempre he recibido su amor, comprensión, paciencia, y quien siempre ha estado incondicionalmente a mi lado para lograr mis metas y objetivos y sobre todo por su gran apoyo, cariño y amor a lo largo de toda mi vida siendo un gran ejemplo para mí y mi formación como persona y profesional.

A mis tíos Bolívar Peralta, Edison Peralta, Cecilia Peralta, Elizabeth Peralta, Beatriz Peralta y a toda mi familia quienes siempre estuvieron en los buenos y malos momentos, por la motivación para cumplir mis objetivos, quienes siempre confiaron en mí y estuvieron allí para darme un consejo o ayudarme incondicionalmente en todo momento y gracias a ellos hoy puedo alcanzar esta meta en mi vida.

A mi enamorada Liliana Chaglla que todo este tiempo de la Universidad siempre estuvo allí para apoyarme en todo momento, enseñando y compartiendo su conocimiento, esto va para la mujer que nunca me dejó rendirme y siempre me dio las fuerzas necesarias para poder cumplir este objetivo tan importante en mi vida. Gracias.

A todos mis amigos que han compartido los buenos y no tan buenos momentos conmigo y que día a día me enseñaron lo que significa la verdadera amistad.

A todos los profesores quienes siempre estuvieron allí, brindándonos siempre su orientación y ayuda en todo momento.



Diego Fernando Quisi Peralta

AGRADECIMIENTOS

A Dios por ser mi guía, compañero darme la vida, a mi madre Luisa Peralta, gracias a todo su esfuerzo, apoyo y confianza que deposito en mí y sobre todo su amor. A toda mi familia especialmente a mis tíos Bolívar, Cecilia, Edison, Elizabeth y Beatriz Peralta quienes me animaron, apoyaron con su gran cariño y siempre estuvieron allí incondicionalmente. Gracias a todos ellos soy quien soy hoy en día.

Agradezco especialmente a mi enamorada y amiga la Ingeniera Liliana Chaglla, quien ha sido un apoyo incondicional en los buenos y malos momentos. Ayudarme en toda la carrera universitaria. Gracias por ser tan especial y única, de verdad que le admiro como persona y también como una gran profesional. Agradezco a Dios por haberte puesto dentro de mi vida. Te Amo Mucho y Gracias por todo.

Mi agradecimiento sincero al Ing. Vladimir Robles Bykbaev. Por su cooperación, dirección, paciencia, entrega y valiosos consejos que me permitieron alcanzar los objetivos de esta tesis y por su gran apoyo incondicional, su conocimiento y experiencia que a lo largo de todo mi carrera me supo transmitir.

Y a todos mis amigos y profesores por su apoyo, ánimo, tiempo y compañía en las diferentes etapas de mi vida. Gracias por formar parte de mi vida, por todo lo que me han brindado y por todas sus bendiciones.



Diego Fernando Quisi Peralta

TABLA DE CONTENIDOS

CAPÍTULO 1	6
1 INTRODUCCIÓN.....	12
1.1 Introducción	12
1.2 Antecedentes.....	13
1.2.1 Justificación	13
1.2.2 Descripción del Problema	13
1.3 Alcance.....	14
1.4 Objetivos	14
1.4.1 Objetivos General.....	14
1.4.2 Objetivos Específicos	15
1.5 Experiencias del Proyecto Pequeñas y Pequeños Científicos de la Universidad Politécnica Salesiana [3].....	15
CAPÍTULO 2	15
2 ESTUDIO DE LA TECNOLOGÍA A DESARROLLAR.....	21
2.1 Estudio de HTML 5 para dispositivos móviles	21
2.2 Estudio del entorno de desarrollo Android	22
2.3 Comparación de la plataforma de desarrollo Android con HTML 5 para dispositivos móviles.....	27
2.3.1 Aplicaciones HTML5 para dispositivos móviles	28
2.3.2 Aplicaciones nativas.....	28
2.4 Especificaciones de la tecnología de desarrollo	30
2.4.1 Instalación y configuración de JDK 7	30
2.4.2 Instalación y configuración del SDK de Android	33
2.4.3 Instalación y configuración de Eclipse Indigo.....	39
2.4.4 Instalar el <i>plugin</i> de Android para Eclipse	39
2.4.5 Creación de un nuevo proyecto Android	41
CAPÍTULO 3	39
3 DISEÑO DEL SOFTWARE.....	45
3.1 Levantamiento de Requerimientos de Software.....	45

3.1.1	Definición de <i>Stakeholders</i>	45
3.1.2	Técnicas de Elicitación [26]	46
3.1.3	Suposiciones y Dependencias.....	46
3.1.4	Requerimientos Funcionales.....	47
3.1.5	Requerimientos no Funcionales.....	50
3.1.6	Restricciones.....	51
3.1.7	Evolución previsible del sistema.....	51
3.1.8	Validación de requerimientos	51
3.2	Diseño de la aplicación.	51
3.2.1	Casos de Uso	51
3.2.2	Diagrama de Clases del Mundo Eléctrico y Verde	53
3.2.3	Diagramas de Secuencia del Mundo Eléctrico y Verde.....	62
3.3	Análisis de las herramientas de desarrollo.	63
3.3.1	Gráficos en Android	63
3.3.2	<i>AsyncTask</i> [27].....	64
3.3.3	Reproducción de sonido [31].....	65
3.3.4	Desarrollo Html5 [23].....	66
CAPÍTULO 4	67
4	IMPLEMENTACIÓN, DOCUMENTACIÓN Y PRUEBAS	73
4.1	Implementación y selección de los módulos de aprendizaje.	73
4.2	Desarrollo de pruebas con los dispositivos móviles en el laboratorio	75
4.3	Desarrollo de pruebas con niños y niñas.	77
4.4	Documentación.	81
4.4.1	Descarga e Instalación de la aplicación.	81
4.4.2	Manual de Usuario	84
4.5	Publicación de la aplicación en la tienda de Google.....	85
5	CONCLUSIONES	95
6	RECOMENDACIONES	97
7	TRABAJOS FUTUROS	97
8	BIBLIOGRAFÍA	98
9	ANEXOS	102
9.1	ANEXO 1: CERTIFICADO DE VALIDACIÓN DE REQUERIMIENTOS.	102
9.2	ANEXO 2: ENCUESTA PRACTICADA A LOS NIÑOS Y NIÑAS	104

ÍNDICE DE FIGURAS

ILUSTRACIÓN 1 PROMEDIO DEL NIVEL DE CONOCIMIENTO DE LOS NIÑOS Y NIÑAS RESPECTO A LOS TEMAS ABARCADOS EN LOS TALLERES, ANTES Y DESPUÉS DE LOS MISMOS.....	16
ILUSTRACIÓN 2 INTERÉS DE LOS NIÑOS Y NIÑAS EN LAS CARRERAS PROFESIONALES PREVIO A LOS TALLERES (PRE-TEST).....	16
ILUSTRACIÓN 3 INTERÉS DE LOS NIÑOS Y NIÑAS EN LAS CARRERAS	16
ILUSTRACIÓN 4 RESPUESTA DE LOS NIÑOS Y NIÑAS A LA PREGUNTA “¿TE GUSTÓ EL TALLER PEQUEÑAS Y PEQUEÑOS CIENTÍFICOS?”	17
ILUSTRACIÓN 5 INTERÉS DE LOS PADRES DE FAMILIA EN QUE SUS NIÑOS Y NIÑAS PARTICIPEN EN TALLERES DE ESTE TIPO.	17
ILUSTRACIÓN 6 TALLERES EN EL AULA DE CIENCIA CON LA ESCUELA FRANCISCA DÁ VILA.	18
ILUSTRACIÓN 7 TALLERES EN EL AULA DE CIENCIA CON LA ESCUELA CARLOS CRESPI.....	18
ILUSTRACIÓN 8 INAUGURACIÓN Y ENTREGA DE DIPLOMAS A 1 ^{ER} GRUPO DE PEQUEÑAS Y PEQUEÑOS CIENTÍFICOS. VISITA DE SECRETARIO DEL SENESCYT, RENÉ RAMÍREZ.	19
ILUSTRACIÓN 9 REUNIÓN CON PADRES DE FAMILIA EN LA ESCUELA CA RLOS CRESPI.	19
ILUSTRACIÓN 10 COMPOSICIÓN DE HTML5 [9].....	22
ILUSTRACIÓN 11 COMPATIBILIDAD DE LOS NA VEGA DORES WEB CON HTML5 [10]	22
ILUSTRACIÓN 12 ARQUITECTURA DE ANDROID	23
ILUSTRACIÓN 13 CICLO DE VIDA DE UN ACTIVITY [20]	26
ILUSTRACIÓN 14 PORCENTAJES DE DISTRIBUCIONES DE ANDROID UTILIZADAS ACTUALMENTE [21]	27
ILUSTRACIÓN 15 SITIO WEB DE ORACLE PARA LA DESCARGA DEL JDK	30
ILUSTRACIÓN 16 DESCARGA DEL JDK.....	31
ILUSTRACIÓN 17 INSTALACIÓN DE JDK 7	31
ILUSTRACIÓN 18CARACTERÍSTICAS DE INSTALACIÓN DEL JDK	32
ILUSTRACIÓN 19 ESTADO DE LA INSTALACIÓN DEL JDK	32
ILUSTRACIÓN 20 SITIO WEB PARA LA DESCARGA DEL SDK DE ANDROID.....	33
ILUSTRACIÓN 21 INSTALACIÓN DEL SDK DE ANDROID	33
ILUSTRACIÓN 22 DIRECTORIO DE INSTALACIÓN DEL SDK DE ANDROID	34
ILUSTRACIÓN 23 PAQUETES DE INSTALACIÓN DE ANDROID	34
ILUSTRACIÓN 24 INSTALACIÓN DE LAS PLATAFORMAS DISPONIBLES DE ANDROID	35
ILUSTRACIÓN 25 ADMINISTRACIÓN DE DISPOSITIVOS VIRTUALES DE ANDROID	36
ILUSTRACIÓN 26 CREACIÓN DE UN NUEVO DISPOSITIVO VIRTUAL DE ANDROID	36
ILUSTRACIÓN 27 CONFIGURACIÓN DE UN NUEVO DISPOSITIVO VIRTUAL DE ANDROID.....	37
ILUSTRACIÓN 28 VISUALIZACIÓN DE LOS DISPOSITIVOS VIRTUALES ANDROID.	38
ILUSTRACIÓN 29 EJECUCIÓN DEL DISPOSITIVO VIRTUAL ANDROID	38
ILUSTRACIÓN 30 SITIO WEB DE DESCARGA DEL ECLIPSE INDIGO 3.7	39
ILUSTRACIÓN 31 INSTALACIÓN DEL PLUGIN ADT EN ECLIPSE.....	40
ILUSTRACIÓN 32 AGREGACIÓN EL REPOSITORIO DE ANDROID	40
ILUSTRACIÓN 33 SELECCIÓN E INSTALACIÓN DE LOS PAQUETES DE ANDROID	41
ILUSTRACIÓN 34 CREACIÓN DE NUEVO PROYECTO DE ANDROID	42

ILUSTRACIÓN 35 CONFIGURACIÓN DE UN NUEVO PROYECTO ANDROID.....	43
ILUSTRACIÓN 36 EJECUCIÓN DE LA APLICACIÓN ANDROID.....	43
ILUSTRACIÓN 37 SITIO WEB DE DESCARGA DE SENCHA ANIMATOR	67
ILUSTRACIÓN 38 INSTALACIÓN DE SENCHA ANIMATOR.....	67
ILUSTRACIÓN 39 DIRECTORIO DE INSTALACIÓN DE SENCHA ANIMATOR.....	68
ILUSTRACIÓN 40 INSTALACIÓN TERMINADA DE SENCHA ANIMATOR	68
ILUSTRACIÓN 41 PANTALLA PRINCIPAL DE SENCHA ANIMATOR	69
ILUSTRACIÓN 42 CONFIGURACIÓN DEL NUEVO PROYECTO DE ANDROID	69
ILUSTRACIÓN 43 PANEL DE OBJETOS DE SENCHA ANIMATOR	70
ILUSTRACIÓN 44 PANEL DE PROPIEDADES DE SENCHA ANIMATOR	70
ILUSTRACIÓN 45 PANEL LÍNEA DE TIEMPO DE SENCHA ANIMATOR	71
ILUSTRACIÓN 46 ESPACIO DE TRABAJO DE SENCHA ANIMATOR	71
ILUSTRACIÓN 47 PANEL DE ESCENAS DE SENCHA ANIMATOR	71
ILUSTRACIÓN 48 PANEL DE CONTROL DE TIEMPO DE SENCHA ANIMATOR	71
ILUSTRACIÓN 49 ENTRADA DE LA DOCTORA ATÓMICA EN EL MUNDO ELÉCTRICO EN 200MS.....	73
ILUSTRACIÓN 50 ENTRADA DE LA DOCTORA ATÓMICA EN EL MUNDO ELÉCTRICO EN 400MS.....	73
ILUSTRACIÓN 51 ENTRADA DE LA DOCTORA ATÓMICA EN EL MUNDO ELÉCTRICO EN 600MS.....	74
ILUSTRACIÓN 52 ENTRADA DEL ÁRBOL CALIPTO EN EL MUNDO VERDE EN 200MS	74
ILUSTRACIÓN 53 ENTRADA DEL ÁRBOL CALIPTO EN EL MUNDO VERDE EN 400MS	75
ILUSTRACIÓN 54 ENTRADA DEL ÁRBOL CALIPTO EN EL MUNDO VERDE EN 600MS	75
ILUSTRACIÓN 55 DISPOSITIVO MÓVIL DE PRUEBA COBY	76
ILUSTRACIÓN 56 DISPOSITIVO MÓVIL DE PRUEBA SAMSUNG GALAXY TAB 2.....	76
ILUSTRACIÓN 57 DISPOSITIVO MÓVIL DE PRUEBAS ASUS TF101	77
ILUSTRACIÓN 58RESULTADO DE LA PREGUNTA ¿CÓMO FUE LA EXPERIENCIA DE UTILIZACIÓN DE LOS TABLETS PARA EL APRENDIZAJE MULTIMEDIA DEL MUNDO VERDE?.....	77
ILUSTRACIÓN 59 RESPUESTA A LA PREGUNTA. ¿AUMENTÓ EL GRADO DE INTERÉS DE APRENDIZAJE UTILIZANDO LOS TABLETS?	78
ILUSTRACIÓN 60 RESPUESTA A LA PREGUNTA. ¿LA UTILIZACIÓN DE LA APLICACIÓN DEL MUNDO VERDE EN LOS TABLETS FUE?.....	78
ILUSTRACIÓN 61 PRUEBA CON NIÑAS DE LA ESCUELA "FRANCISCA DÁ VILA"	79
ILUSTRACIÓN 62 PRUEBA CON LAS NIÑAS E INSTRUCTORES.....	79
ILUSTRACIÓN 63 PRUEBA CON UN NIÑO DE LA ESCUELA "CARLOS CRESPI" CON EL TABLET COBY	80
ILUSTRACIÓN 64 PRUEBAS CON NIÑOS DE LA ESCUELA "CARLOS CRESPI"	80
ILUSTRACIÓN 65 PRUEBA CON UN ESTUDIANTE DE LA ESCUELA “CARLOS CRESPI” CON LA TABLET SAMSUNG GALAXY TAB 2.....	81
ILUSTRACIÓN 66 GESTOR DE DESCARGAS DE APLICACIONES PLAY STORE [35].	81
ILUSTRACIÓN 67 BÚSQUEDA DE LA APLICACIÓN EN PLA Y STORE	82
ILUSTRACIÓN 68 DESCRIPCIÓN DE LA APLICACIÓN DEL MUNDO VERDE	82
ILUSTRACIÓN 69 ACEPTAR Y DESCARGAR LA APLICACIÓN DEL MUNDO VERDE	83
ILUSTRACIÓN 70 INSTALACIÓN COMPLETA DE LA APLICACIÓN MUNDO VERDE.....	83
ILUSTRACIÓN 71 ICONO DE LA APLICACIÓN DEL MUNDO VERDE	84
ILUSTRACIÓN 72 MENÚ PRINCIPAL DEL MUNDO VERDE	84
ILUSTRACIÓN 73 MENÚ DE SUBTEMAS DEL MUNDO VERDE	84
ILUSTRACIÓN 74 TRAMOS DEL MUNDO VERDE	85

ILUSTRACIÓN 75 CREACIÓN DE UNA CUENTA EN GMAIL	86
ILUSTRACIÓN 76 CONDICIONES DE SERVICIOS Y POLÍTICAS DE SEGURIDAD DE GOOGLE ...	86
ILUSTRACIÓN 77 PANTALLA DE BIENVENIDA A GMAIL	87
ILUSTRACIÓN 78 CREACIÓN DEL PERFIL DESARROLLADOR EN <i>GOOGLE PLAY</i>	87
ILUSTRACIÓN 79 ACUERDO DE DISTRIBUCIÓN PARA DESARROLLADORES	88
ILUSTRACIÓN 80 PAGO DE CUOTA DE REGISTRO	88
ILUSTRACIÓN 81 PANEL DE ADMINISTRACIÓN DE PUBLICACIONES EN ANDROID	89
ILUSTRACIÓN 82 VENTANA PARA SUBIR EL PROYECTO DE ANDROID	89
ILUSTRACIÓN 83 SELECCIÓN DEL .APK	89
ILUSTRACIÓN 84 CUADRO DE PROGRESO DE SUBIR LA APLICACIÓN AL SERVIDOR.....	90
ILUSTRACIÓN 85 INFORMACIÓN DE LA APLICACIÓN SUBIDA AL SERVIDOR	90
ILUSTRACIÓN 86 ESPECIFICACIONES DE LA INFORMACIÓN DEL MUNDO VERDE	91
ILUSTRACIÓN 87 ESPECIFICACIONES GRÁFICAS DEL MUNDO VERDE	92
ILUSTRACIÓN 88 CATEGORIZACIÓN Y DETALLE DE CONTACTO DEL MUNDO VERDE	92
ILUSTRACIÓN 89 ESPECIFICACIÓN DEL PRECIO Y DISTRIBUCIÓN DEL MUNDO VERDE.....	93
ILUSTRACIÓN 90 PRESENTACIÓN DE LA APLICACIÓN EN <i>GOOGLE PLAY</i> DEL MUNDO VERDE.....	94

ÍNDICE DE TABLAS

TABLA 1 TALLERES REALIZADOS POR LOS NIÑAS Y NIÑOS EN EL AULA DE PEQUEÑAS Y PEQUEÑOS CIENTÍFICOS.....	16
TABLA 2 MATRIZ DE LOS STAKEHOLDERS	45

CAPÍTULO 1

1 INTRODUCCIÓN

1.1 Introducción

Las Tecnologías de la Información y la Comunicación TIC's están sufriendo un desarrollo vertiginoso, lo cual está afectando a prácticamente todos los campos de nuestra sociedad, y la educación no es una excepción [1], lo que conlleva a que las tecnologías de información lleguen a ser un pilar básico en la sociedad. Por ello, los gobiernos están invirtiendo en su desarrollo, de modo que las escuelas, colegios y universidades puedan emplear las TICS como una herramienta de soporte en la enseñanza.

Actualmente las TICS ya están cambiando ciertos procesos educativos mediante el internet, como por ejemplo, educación a distancia, el auto aprendizaje, cursos en línea, video conferencia, etc., lo que produce que sea un complemento o enriquecimiento de la enseñanza tradicional. Otro campo de utilización lo constituye el apoyo en el aprendizaje, en donde las TICS son integradas pedagógicamente dentro del proceso de enseñanza del aula, como por ejemplo, el uso de dispositivos móviles para un aprendizaje didáctico [4].

Las TICS están jugando un papel muy importante en el desarrollo de los países, siendo un factor importante para que los mismos puedan llegar a alcanzar ventajas competitivas frente a los demás. El Ecuador no está dentro de países que lideran propuestas de innovación en políticas y regulaciones que buscan desarrollar y asimilar estas tecnologías. Esto produce que en el país los estudiantes, entre una de sus causas, no se enfoquen al ámbito de la investigación.

En este mundo globalizado, para poder lograr la inclusión de las TICS, es importante incorporarlas en los entornos educativos y aplicar su uso, a través del uso de los dispositivos móviles para el aprendizaje o educación virtual en los niños.

Una posible solución a esta problemática es que se empiece a inculcar la auto-investigación desde temprana edad, motivando esencialmente al estudio de las ciencias exactas. Esto permitirá crear en los niños y niñas un interés por la tecnología e indirectamente afectar de manera positiva en el desarrollo de la sociedad. La herramienta propuesta pretende complementar la labor de enseñanza de distintas ciencias por parte de los docentes escolares [3]. Consideramos que esta enseñanza se debe realizar con el apoyo de la tecnología de los dispositivos móviles, lo que probablemente ayude a motivar más a los niños y niñas en el uso de las tecnologías de la información, auto-aprendizaje e investigación.

Actualmente ya existe un proyecto en la Universidad Politécnica Salesiana para apoyar el aprendizaje de los niños y niñas, en lo que podemos apreciar como el aprendizaje de las ciencias exactas con ayuda de las tics pueden influenciar positivamente tanto en el desarrollo intelectual como creativo y emprendedor de los niños y niñas.

Las tecnologías de información se está renovando continuamente por lo que se ha visto la posibilidad de desarrollar la aplicación para los dispositivos móviles utilizando tecnologías web como HTML5, CSS3 y JavaScript, y empaquetar dentro de una aplicación nativa para la visualización, obteniendo como resultado aplicaciones híbridas potencializando las dos tecnologías, sin más preámbulo se procederá al desarrollo del mismo.

1.2 Antecedentes

1.2.1 Justificación

Actualmente la Universidad Politécnica Salesiana está desarrollando un proyecto para apoyar el aprendizaje de niños y niñas y despertar su interés por el estudio de la ciencia y la tecnología, todo esto desde una perspectiva basada en la experimentación y guías multimedia de enseñanza. Este proyecto ya cuenta con un portal web (www.pequeciencia.ups.edu.ec), sin embargo, se está renovando y dentro de este proceso se ha visto la necesidad de incluir la plataforma de enseñanza en dispositivos móviles, ya que el continuo avance tecnológico de éstos ha permitido que los usuarios puedan contar con herramientas de acceso virtual a diversos procesos educativos.

En la actualidad las empresas están analizando la posibilidad de trasladar sus aplicaciones a los dispositivos móviles como una estrategia en el mercado muy sofisticada, ya que el usuario se identifica con los aplicativos de la empresa, consecuentemente las empresas están buscando tener presencia en estos dispositivos.

El beneficio que la Universidad Politécnica Salesiana obtendrá, es mejorar sus procesos de vinculación con la colectividad y ayudar a que los niños y niñas del Azuay y el Ecuador puedan ser motivados por la tecnología, despertando en ellos y ellas ese amor por la ciencia.

El software desarrollado podrá ser descargado en cualquier parte del mundo de forma gratuita, llevando consigo el aval de la Universidad Politécnica Salesiana y del Ecuador. Este programa permitirá fomentar la investigación a través del uso de aplicativos de dispositivos móviles y la riqueza técnica generada puede servir de base para en un futuro tener mejores procesos de estimulación educativa en niños.

1.2.2 Descripción del Problema

Como se mencionó anteriormente, el proyecto Pequeñas y Pequeños Científicos ya posee un portal web, en donde los niños y niñas pueden profundizar su aprendizaje sobre los mundos digital¹, eléctrico², magnético³ y verde⁴, pero una gran desventaja

¹Mundo Digital: Representa la evolución de la tecnología impartiendo temas como el funcionamiento de los aparatos digitales, trabajar con datos binarios y por ultimo una breve introducción a la programación de software.

es que ellos deben tener un dispositivo con acceso a internet para poder utilizar éstos servicios. En virtud de ello, se busca contar con esta aplicación en dispositivos móviles, preferentemente en *tablets*. De esta forma, los niños y las niñas se podrán descargar completamente la aplicación y tendrán la posibilidad de involucrarse en este mundo tecnológico.

Actualmente el continuo avance de la tecnología pone a disposición de los usuarios una gran variedad de dispositivos móviles, los cuales ofrecen servicios que agilitan los procesos diarios, ofreciendo una amplia gama de aplicaciones para todo propósito. Un ejemplo de estos dispositivos son los *tablets*, los cuales se asemejan a un mini laptop del tamaño de un cuaderno pequeño que no necesita de periféricos para su utilización, ya que incorpora la tecnología *touchscreen*⁵[2] a un bajo costo.

Uno de los principales motivos por los que se ha desarrollado en la tecnología Android, es que el 68.1% de los dispositivos móviles a nivel mundial usan Android como sistema operativo[5] y por otro lado es completamente gratuito, por lo que existe un gran número de fabricantes de dispositivos que están utilizando esta tecnología, de forma que el número de usuarios está creciendo exponencialmente.

1.3 Alcance

Se considera dentro del alcance de este proyecto:

- Que la aplicación Android del proyecto Pequeñas y Pequeños Científicos funcione correctamente en dispositivos con esta tecnología.
- Que la aplicación desarrollada se cargue a la tienda de *GooglePlay* para su publicación y posterior descarga gratuita.
- Creaciones de animaciones y guías multimedia utilizando Android para un mejor aprendizaje de los niños.
- Redacción de la memoria del proyecto

1.4 Objetivos

1.4.1 Objetivos General

Diseñar e implementar una aplicación para dispositivos Android para en el marco del proyecto “Pequeñas y Pequeños Científicos de la Universidad Politécnica Salesiana”.

²Mundo eléctrico: Representa la utilización de la energía para todos los dispositivos digitales, dando a conocer temas como la historia de la electricidad, tipos de electricidad y por último el circuito eléctrico.

³Mundo Magnético: Representa la utilización de imanes impartiendo temas como el magnetismo, tipos de imanes, electromagnetismo y por último el motor eléctrico.

⁴Mundo Verde: Representa la generación de energía y el medio ambiente impartiendo temas de vital importancia como la contaminación, energías renovables y por último el cuidado del medio ambiente.

⁵Permite que mediante el toque directo sobre la pantalla la entrada de los datos y órdenes al dispositivo, y a su vez muestra los resultados en la pantalla, actuando como periférico de entrada y salida de datos.

1.4.2 Objetivos Específicos

- Desarrollar una aplicación Android para el proyecto Pequeñas y Pequeños Científicos que estará conformada por uno o más módulos de aprendizaje de los mundos digital, verde, eléctrico y magnético.
- Generar una base de conocimiento debidamente documentada sobre el desarrollo de proyectos Android
- Realizar pruebas en diferentes tipos de dispositivos móviles (preferentemente entablets).
- Determinar cuáles son los requerimientos mínimos para que las aplicaciones se ejecuten sin ningún problema en dispositivos basados en Android.
- Cargar la aplicación terminada a la tienda de Google (*Google Play*).
- Estudiar la tecnología Android para el desarrollo de aplicaciones móviles y de su metodología para la documentación

1.5 Experiencias del Proyecto Pequeñas y Pequeños Científicos de la Universidad Politécnica Salesiana [3]

Desde el mes de diciembre del 2011 se han venido realizando diversas actividades enfocadas a la implementación del Aula de Ciencia de “Pequeñas y Pequeños Científicos”. Se desarrollaron 8 talleres de ciencia, abarcando un total de 50 niños y 47 niñas con un total de 97 estudiantes llegando a firmar convenios con las escuelas “Carlos Crespi” y “Francisca Dávila” de la ciudad de Cuenca, en la siguiente tabla se detalla cada uno de los talleres dictados.

TALLERES REALIZADOS							
Taller	Escuela	Cupo	Fecha inicio	de	Fecha finalización	de	Horario
Taller 1	CARLOS CRESPI	12 niños	Lunes 9 de abril		Viernes 20 de abril		15h00-16h00
Taller 2	FRANCISCA DÁVILA	12 niñas	Lunes 9 de abril		Viernes 20 de abril		16h00-17h00
Taller 3	CARLOS CRESPI	12 niños	Lunes 23 de abril		Viernes 4 de mayo		15h00-16h00
Taller 4	FRANCISCA DÁVILA	12 niñas	Lunes 23 de abril		Viernes 4 de mayo		16h00-17h00
Taller 5	CARLOS CRESPI	12 niños	Lunes 7 de mayo		Viernes 18 de mayo		15h00-16h00
Taller 6	FRANCISCA DÁVILA	12 niñas	Lunes 7 de mayo		Viernes 18 de mayo		16h00-17h00
Taller 7	CARLOS CRESPI	12 niños	Lunes 21 de mayo		Viernes 1 de junio		15h00-16h00

Taller 8	FRANCISCA DÁVILA	12 niñas	Lunes 21 de mayo	Viernes 1 de junio	16h00-17h00
-----------------	---------------------	----------	------------------	--------------------	-------------

Tabla 1 Talleres realizados por Niñas y Niños en el Aula de Pequeñas y Pequeños Científicos.

Resultados del proyecto:

- Resultados académicos.

Se ha observado un incremento del 48% en el promedio del nivel del conocimiento de los niños y niñas de los temas abarcados. Estos resultados se han obtenido de la comparación entre el pre-test y el post-test. La metodología utilizada para alcanzar este resultado no contempló la necesidad de enviar deberes, trabajos, ni estudiar en horas fuera del taller.

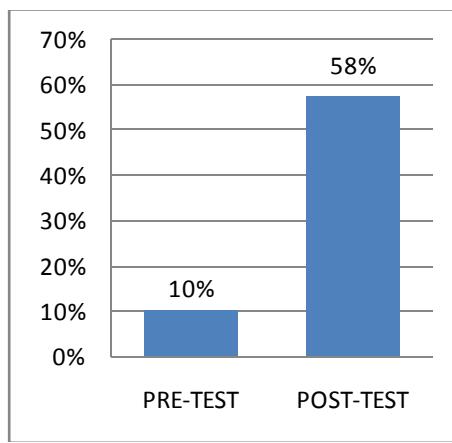


Ilustración 1 Promedio del nivel de conocimiento de los niños y niñas respecto a los temas abarcados en los talleres, antes y después de los mismos.

Se planteó la pregunta: “¿Qué deseas ser cuando seas grande?”. En el pre-test apenas el 31% de los niños y niñas se inclinaban por carreras vinculadas con la ciencia y la ingeniería. Despues de los talleres el 65% expresó que su deseo era convertirse en científico o ingeniero.

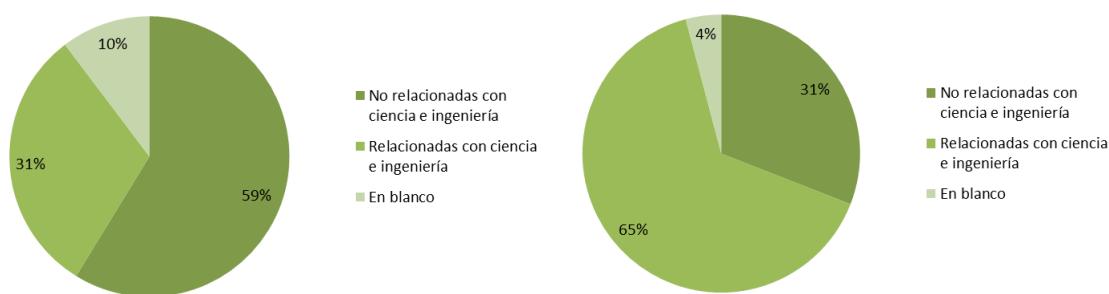


Ilustración 2 Interés de los niños y niñas en las carreras profesionales previo a los talleres (pre-test).

Ilustración 3 Interés de los niños y niñas en las carreras

El 98% de los estudiantes manifestó su gran satisfacción de haber asistido a los talleres “Pequeñas y Pequeños Científicos”. La razón predominante de este gusto es esencialmente la metodología de enseñanza que se aplicó, basada en adquirir nuevos conocimientos de una manera divertida, por medio de la experimentación y el juego.

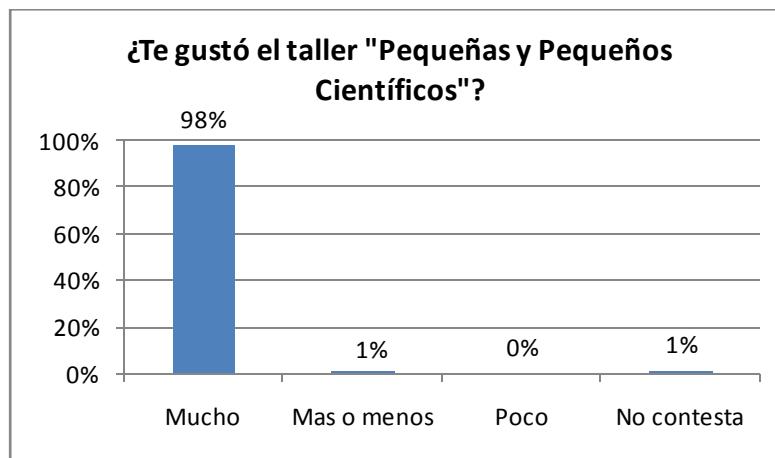


Ilustración 4 Respuesta de los niños y niñas a la pregunta “¿Te gustó el taller Pequeñas y Pequeños Científicos?”

- Impacto en los padres de familia.

Se aplicó la encuesta a los padres de familia a partir de la cual se determinó que al 96% de los mismos les gustaría mucho que sus hijos e hijas participen en este tipo de talleres.

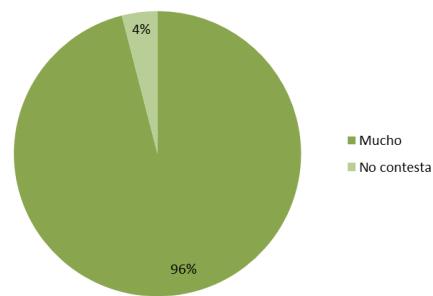


Ilustración 5 Interés de los padres de familia en que sus niños y niñas participen en talleres de este tipo.

El 97% de los padres de familia expresaron que sus hijos e hijas diariamente sentían muchas ganas de asistir a los talleres de ciencia. Asimismo, el 95% dijo que sus hijos e hijas realizaban comentarios positivos en casa acerca de los talleres.

La repercusión de los talleres de ciencia expresa que el 71% de los padres de familia percibieron que sus hijos e hijas, después de los talleres, se interesaban mucho por noticias de ciencia y tecnología, y asimismo, un 75% dijo que sus niños y niñas se sentían muy motivados en conocer el porqué de las cosas y cómo funcionan.

- Impacto en los centros educativos.

El proyecto ha tenido una excelente acogida por parte los centros educativos con los cuales se trabajó durante esta primera etapa, quienes han manifestado que esto ha sido una experiencia enriquecedora en el desarrollo intelectual, creativo y emprendedor de los niños y niñas participantes.

A continuación se muestra unas imágenes de cómo los niños y niñas toman las clases, se entregan los respectivos diplomas por haber cruzado con éxito el taller y como sus padres son informados de los avances que tiene sus hijos del proyecto Pequeñas y Pequeños Científicos.



Ilustración 6 Talleres en el Aula de ciencia con la escuela Francisca Dávila.



Ilustración 7 Talleres en el Aula de ciencia con la escuela Carlos Crespi.



Ilustración 8 Inauguración y entrega de diplomas a 1^{er} grupo de Pequeñas y Pequeños Científicos. Visita de Secretario del Senescyt, René Ramírez.



Ilustración 9 Reunión con padres de familia en la escuela Carlos Crespi.

CAPÍTULO 2

2 ESTUDIO DE LA TECNOLOGÍA A DESARROLLAR

2.1 Estudio de HTML 5 para dispositivos móviles

La aparición de internet y del *World Wide Web*, ha cambiado para siempre nuestra forma de acceder a la información. Con la progresiva conversión de la Web en la conocida Web 2.0, los usuarios pasamos de ser consumidores de información a generadores de contenido, por consiguiente manejar y entender el lenguaje HTML5, CSS3 y JavaScript nos permitirá controlar a la perfección los materiales que deseemos publicar en internet, proporcionándoles la forma y la estructura apropiada[36].

“HTML5 es un estándar que empezó a desarrollarse en el año 2009, y que nos ofrece nuevas características para lograr de forma más eficiente funciones que antes sólo se lograban instalando complementos como Flash Player o que sólo eran posibles en aplicaciones de escritorio o *apps*⁶ nativas para los distintos dispositivos móviles y *tablets*, pero se sigue conservando etiquetas de sus antecesores HTML4 y XHTML” [7].

HTML5 ya que no es simplemente una nueva versión del lenguaje de marcación HTML, sino una agrupación de diversas especificaciones concernientes al desarrollo web, se pretende proporcionar una plataforma con la que se desarrolle aplicaciones web más parecidas a las aplicaciones de escritorio. Para ello se están desarrollando *APIS*⁷ que permitan trabajar con cualquiera de los elementos de la página y realizar acciones que hasta hoy era necesario realizar por medio de tecnologías adicionales [6], además, por su características de código libre, renderizable⁸ desde el navegador, flexibilidad y ligereza, hace que HTML5 sea una gran opción para los dispositivos móviles [8].

Las reglas básicas que se plantearon a la hora de actualizar a HTML 5 fueron [11]:

- Basarse en HTML, CSS, DOM y JavaScript.
- Reducir la necesidad del uso de *plugins*⁹, como por ejemplo flash.
- Mejorar el tratamiento de errores.
- Crear etiquetas que reemplacen el uso de scripts.
- Lenguaje que pueda utilizarse en cualquier tipo de dispositivo, como móviles, *tablets*, etc.

⁶Apps es una abreviatura de aplicación que puede funcionar en internet, en el equipo o en un dispositivo móvil.

⁷Apis es una abreviación de Interfaz de programación de aplicaciones, es un conjunto de rutinas, protocolos y herramientas para la construcción de aplicaciones de software.

⁸La renderización se encarga de interpretar las marcas (HTML, XML, imágenes, etc.) o la información de formato (CSS, XSL, etc.), para mostrar el contenido formateado, es decir, con las formas y estilos apropiados[32].

⁹Es un complemento de una aplicación que se relaciona con otra, para aportarle una función nueva y generalmente muy específica.

- Eliminación de etiquetas y atributos obsoletos o no deseables.

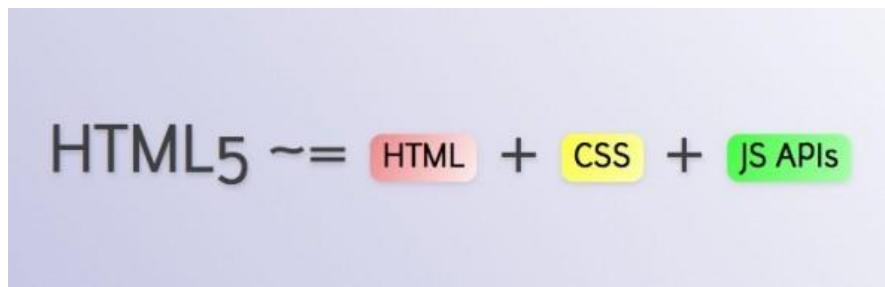


Ilustración 10 Composición de Html5 [9]

Sin embargo, el principal problema es que aún se está definiendo el lenguaje. Por ello, al no encontrarse estandarizado, los navegadores aún no soportan al 100% sus nuevas funcionalidades [34]. A continuación se puede apreciar el grado de compatibilidad de los principales navegadores web.

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
		3.6: 45%						10.0: 35%	2.1: 23%
	6.0: 15%	9.0: 70%				3.2: 24%		11.0: 56%	2.2: 34%
	7.0: 15%	10.0: 70%	17.0: 81%			4.0-4.1: 34%		11.1: 63%	2.3: 38%
	8.0: 18%	11.0: 70%	18.0: 82%	5.0: 63%		4.2-4.3: 40%		11.5: 63%	3.0: 55%
Current	9.0: 42%	12.0: 70%	19.0: 82%	5.1: 73%	11.6: 70%	5.0: 68%	5.0-6.0: 11%	12.0: 73%	4.0: 61%
Near future	10.0: 70%	13.0: 70%	20.0: 88%	5.2: 79%	12.0: 79%				
Farther future		14.0: 70%	21.0: 95%						

Ilustración 11 Compatibilidad de los navegadores web con html5 [10]

“HTML5 permiten una flexibilidad y potencia muy grandes para la creación de animaciones, además de hacerlas independientes del dispositivo en el que se está visualizando: ordenadores, móviles, iPads, tablets, etc.” [18], ejecutadas dentro de un componente WebKit del móvil.

2.2 Estudio del entorno de desarrollo Android¹⁰

Android es una plataforma para dispositivos móviles que contiene una fuente de software donde se incluye un sistema operativo, middleware¹¹ y aplicaciones básicas para el usuario, con las siguientes características:

- Desarrollo rápido de aplicaciones, que sean reutilizables y verdaderamente portables entre diferentes dispositivos.

¹⁰Toda ésta sección está tomada textual e íntegramente de [14].

¹¹Middleware es un software de computadora que conecta componentes de software o aplicaciones para que puedan intercambiar datos o comunicarse entre éstas [19].

- Cuenta con su propia máquina virtual, *Dalvik*, que interpreta y ejecuta código escrito en Java.
- Permite la representación de gráficos 2D y 3D.
- Posibilita el uso de bases de datos.
- Soporta un elevado número de formatos multimedia.
- Servicio de localización GSM.
- Controla los diferentes elementos hardware: Bluetooth, Wi-Fi, cámara fotográfica o de vídeo, GPS, acelerómetro, infrarrojos, etc.
- Cuenta con un entorno de desarrollo muy cuidado mediante un SDK disponible de forma gratuita.
- Ofrece un *plugin* para uno de los entornos de desarrollo más populares, Eclipse, y un emulador integrado para ejecutar las aplicaciones

Un dispositivo Android cuenta con la siguiente arquitectura (ilustración 12), la misma que se procederá a detallar sus componentes más importantes.



Ilustración 12 Arquitectura de Android

La capa más inmediata es la que corresponde al núcleo de Android, utiliza el núcleo de Linux 2.6 como una capa de abstracción para el hardware disponible en los dispositivos móviles, la cual contiene los drivers necesarios para cualquier componente de hardware pueda ser utilizado. Cabe recalcar que el fabricante es el encargado de crear las correspondientes librerías de control *odrivers*.

La siguiente capa corresponde con las librerías utilizadas por Android, estas han sido escritas utilizando C/C++ y proporcionan a Android la mayor parte de sus capacidades y características:

- La librería *libc* incluye todas las cabeceras y funciones según el estándar del lenguaje C. Todas las demás librerías se definen en este lenguaje.
- La librería *SurfaceManager* es la encargada de componer los diferentes elementos de navegación de pantalla. Gestiona también las ventanas pertenecientes a las distintas aplicaciones activas en cada momento.
- *OpenGL/SL* y *SGL* representan las librerías gráficas y por tanto sustentan la capacidad gráfica de Android.
- La librería *MediaPlayer* proporciona todos los *códecs* necesarios para el contenido multimedia soportado en Android (vídeo, audio, imágenes estáticas y animadas, etc.)
- A través de la librería *SQLite*, Android ofrece la creación y gestión de bases de datos relacionales.
- La librería *WebKit* proporciona un motor para las aplicaciones de tipo navegador, y forma el núcleo del actual navegador incluido por defecto en la plataforma Android.

En este nivel también se encuentran las librerías de Android, entre las cuales encontramos las *CoreLibraries*, estas están desarrolladas en lenguaje java, y la maquina virtual *Dalvik*, que constituyen el framework de aplicaciones el cual representa fundamentalmente el conjunto de herramientas de cualquier aplicación.

Dentro de este framework de aplicaciones podemos mencionar algunas de las librerías más importantes:

- *Activity Manager*: Importante conjunto de *APIs* que gestiona el ciclo de vida de las aplicaciones en Android.
- *Window Manager*: Gestiona las ventanas de las aplicaciones
- *TelephoneManager*: Incluye todas las *APIs* vinculadas a las funcionalidades propias del teléfono (llamadas, mensajes, etc.)
- *Content Providers*: Permite a cualquier aplicación compartir sus datos con las demás aplicaciones de Android. Por ejemplo, gracias a esta *API* la información de contactos, agenda, mensajes, etc. será accesible para otras aplicaciones.
- *View System*: Proporciona un gran número de elementos para poder construir interfaces de usuario (*GUI*), como listas, mosaicos, botones, *check-boxes*, tamaño de ventanas, control de las interfaces mediante tacto o teclado, etc. Incluye también algunas vistas estándar para las funcionalidades más frecuentes.

- *Location Manager*: Posibilita a las aplicaciones la obtención de información de localización y posicionamiento.
- *Notification Manager*: Permite que las aplicaciones usen un mismo formato para comunicar al usuario eventos que ocurrán durante su ejecución, por ejemplo, una llamada entrante, un mensaje recibido, conexión Wi-Fi disponible, ubicación en un punto determinado, etc.

La máquina virtual *Dalvik* ha sido optimizada y adaptada a las peculiaridades propias de los dispositivos móviles (menor capacidad de proceso, baja memoria, alimentación por batería, etc.) y trabajar con ficheros de extensión .dex(*Dalvik Executables*). *Dalvik* no trabaja directamente con el *bytecode* de Java, sino que lo transforma en un código más eficiente que el original, pensado para procesadores pequeños.

Los ficheros .class de Java se compilan en ficheros .dex, de forma que cada fichero .dex puede contener varias clases. Después, este resultado se comprime en un único archivo de extensión .apk(*Android Package*), el cual es el que se distribuirá a los dispositivos móviles.

Una vez vista la arquitectura de Android, debemos revisar cuáles son los componentes básicos de una aplicación:

- *Activity*: Un componente *Activity* refleja una determinada actividad llevada a cabo por una aplicación y se asocia típicamente a una ventana o interfaz de usuario.
- *BroadcastIntent Receiver*: Se utiliza para lanzar una ejecución dentro de la aplicación actual cuando un determinado evento se produzca. El sistema lanzará la aplicación si es necesario cuando el evento monitorizado tenga lugar.
- *Service*: Representa una aplicación ejecutada sin interfaz de usuario y que generalmente tiene lugar en segundo plano mientras otras aplicaciones son las que están activas en la pantalla del dispositivo.
- *Content Provider*: Una clase que implemente este componente contendrá una serie de métodos que permite almacenar, recuperar, actualizar y compartir los datos de una aplicación ya sea en archivos o la base de datos *SQLite*.

Cada uno de los componentes básicos de Android tiene un ciclo de vida bien definido; esto implica que el desarrollador puede controlar en cada momento en qué estado se encuentra dicho componente, pudiendo así programar las acciones.

El componente *Activity*, probablemente el más importante tiene el siguiente ciclo de vida:

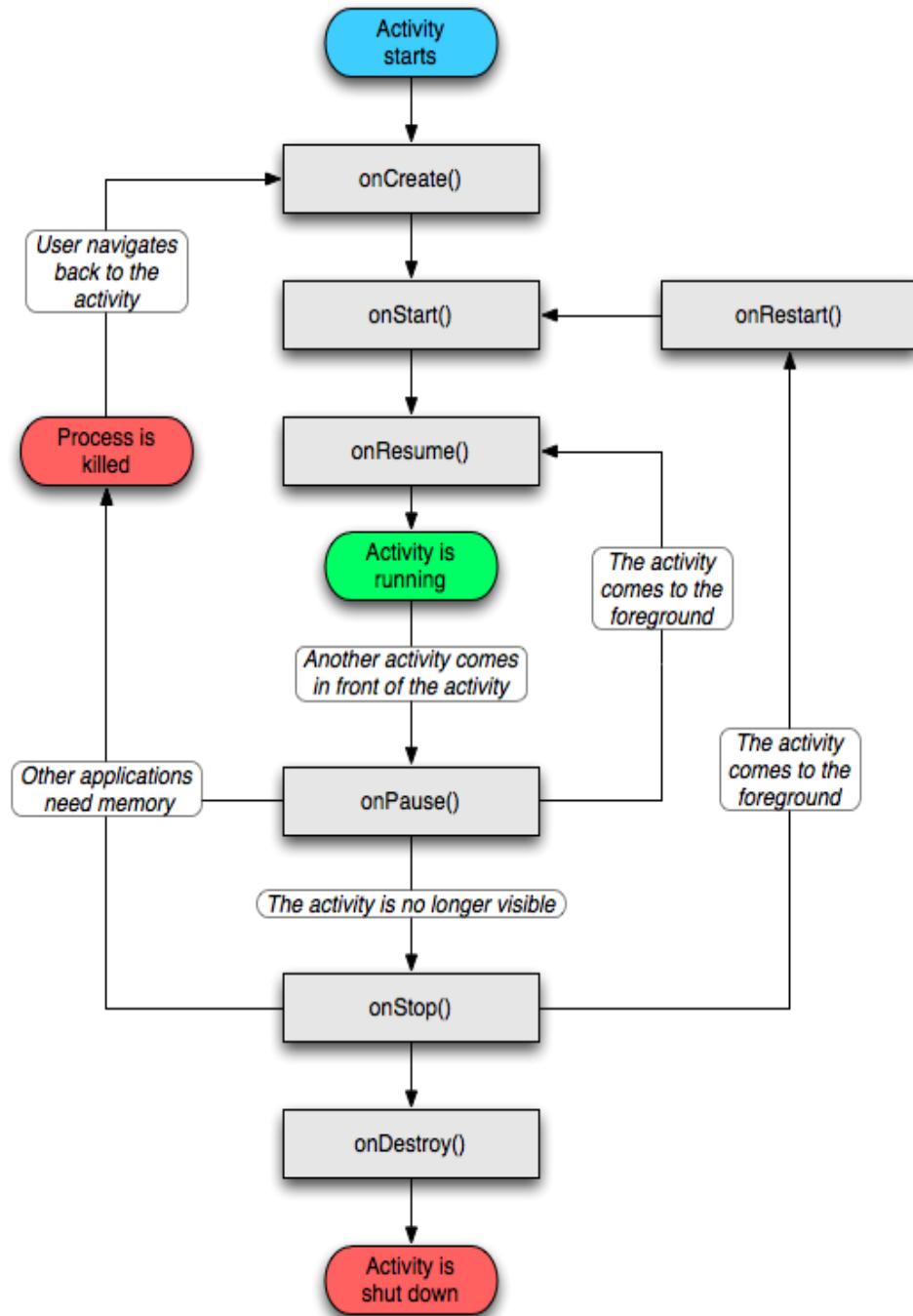


Ilustración 13 Ciclo de vida de un *Activity* [20]

En la ilustración 13 podemos observar que entre los principales eventos que tenemos son:

- *onCreate()*, *onDestroy()*: Abarcan todo el ciclo de vida. Cada uno de estos métodos representan el principio y el fin de la actividad.

- *onStart(), onStop()*: Representan la parte visible del ciclo de vida. Desde *onStart()* hasta *onStop()*, la actividad será visible para el usuario.
- *onResume(), onPause()*: Delimitan la parte útil del ciclo de vida. Desde *onResume()* hasta *onPause()*, la actividad no es visible.

La mayoría de las medidas de seguridad entre el sistema y las aplicaciones deriva de los estándares de Linux 2.6. Por defecto, ninguna aplicación tiene permiso para realizar ninguna operación o comportamiento que pueda impactar negativamente en la ejecución de otras aplicaciones o del sistema operativo. La única forma de poder saltar estas restricciones impuestas por Android, es mediante la declaración explícita de un permiso que autorice a llevar a cabo una determinada acción.

Android tiene varios avances en tecnología y funcionalidad, cada uno de estos avances conlleva a una actualización en las cuales se arreglan los problemas y se agregan nuevas funciones. Actualmente Android se encuentra en la versión 4.1 (*Jelly Bean*), en la ilustración 14 se puede apreciar las versiones de Android más utilizadas en los celulares y *tablets*, realizada en noviembre del 2012

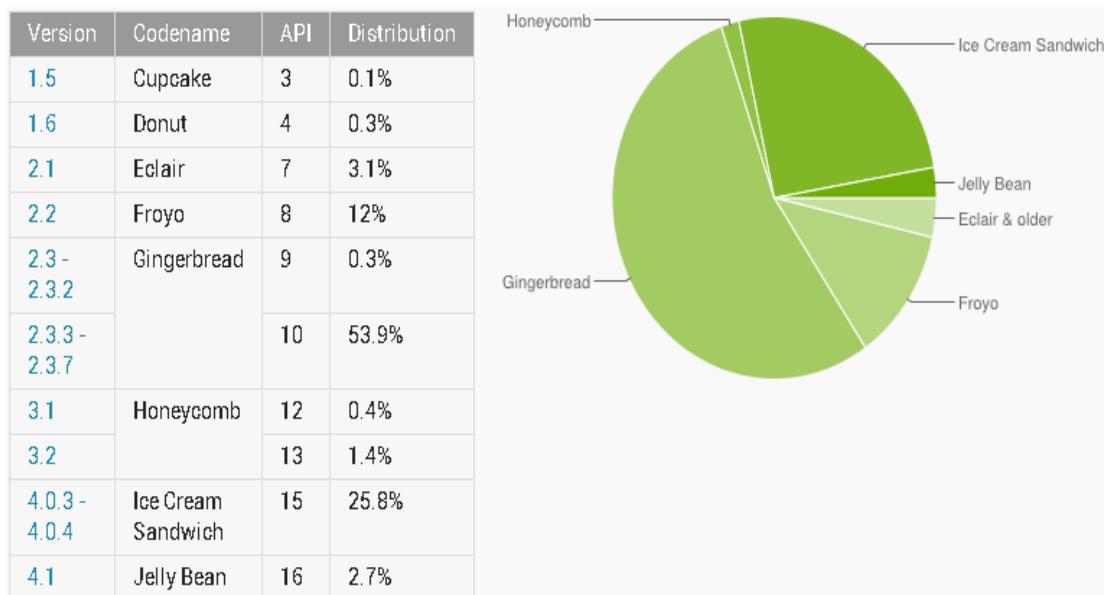


Ilustración 14 Porcentajes de distribuciones de Android utilizadas actualmente [21]

2.3 Comparación de la plataforma de desarrollo Android con HTML 5 para dispositivos móviles.

Se procederá a detallar las ventajas y desventajas de las dos plataformas de desarrollo para dispositivos móviles.

2.3.1 AplicacionesHTML5 para dispositivos móviles.

Una de las principales ventajas del desarrollo en dispositivo móviles es que todos los dispositivos tienen un navegador web, que ya incorpora la tecnología HTML5, por lo que podemos crear una aplicación web y desplegar la aplicación dentro de un visor Web como si fuera nativa [15].

Ventajas [15]:

- Es de fácil diseño ya que no siempre se debe desarrollar una aplicación para móviles. Se puede realizar un diseño adaptado a una pantalla y se podrá visualizar como una aplicación nativa.
- Fácil implementación en las aplicaciones que se pueden desarrollar utilizando tecnologías web como HTML5, CSS3 y JavaScript, permitiendo así optimizar el tiempo de desarrollo.
- Manejo de seguridad utilizando librerías de JavaScript.
- Acceso al hardware del dispositivo lo que permite acceder a las características del dispositivo móvil como por ejemplo el GPS, acelerómetro, cámara, contactos, base de datos, *filesystem*, etc., por medio de un *framework*[13].
- Es la solución que más plataformas móviles soporta, ya que corre dentro de un navegador web. Además de Iphone/Ipad y Android, funciona también en Palm, Symbian, WebOS, Windows 7 y BlackBerry [13].
- Hay buena documentación y bastantes ejemplos en línea permitiendo un mejor desarrollo [13].
- La licencia es gratuita [13].

Desventajas

- Mayor consumo de recursos ya que se debe renderizar HTML e interpretar JavaScript, lo cual es más costoso que ejecutar una aplicación nativa en términos de recursos de hardware del dispositivo[15].

2.3.2 Aplicaciones nativas.

Entre las principales ventajas y desventajas del desarrollo nativo en dispositivos móviles basados en Android son los siguientes:

Ventajas [22]:

- Se puede instalar en casi todo tipo de dispositivos, lo cual permite que los diferentes fabricantes y operadoras tomen a este sistema operativo en sus equipos [16].

- Android tiene un sistema completamente libre para que un desarrollador no solo pueda modificarlos sino también mejorarlo.
- Entorno de desarrollo libre basado en Eclipse y compatible con varios sistemas operativos, que cuenta con simulador de Android para probar las aplicaciones.
- Librerías y componentes reutilizables para todo tipo de aplicaciones.
- Gran comunidad de desarrolladores, variasfuentes de información acerca de la plataforma con ejemplos de cada una de las aplicaciones.
- Más de 100.000 aplicaciones disponibles para teléfonos Android la mayoría gratis.
- Es un sistema operativo multitarea que es capaz de gestionar varias aplicaciones abiertas a la vez, dejando en suspensión aquellas que no se utilicen y cerrarlas en caso de resultar ya inútiles para evitar un consumo de memoria[16].

Desventajas [22]:

- Google tiene una política restrictiva hacia las versiones más recientes, no haciéndolas públicas hasta que ellos lo vean conveniente.
- Como Android es un sistema operativo de código abierto produce que este expuesto a vulnerabilidades.

Después de un análisis podemos llegar a la conclusión de que la tecnología web es un complemento del desarrollo para dispositivos móviles en su lenguaje nativo, ya que nos permite desarrollar bajo una tecnología utilizando HTML5 con todas sus potencialidades. Por otra parte, CSS3 nos ayuda a la realización de animaciones, estilos y efectos en los componentes de HTML y la potencialidad de JavaScript para el manejo de eventos, servicios web, accesos a las características del dispositivo como la cámara, GPS, acelerómetros, etc. A este tipo de aplicaciones se les conoce como aplicaciones híbridas [15].

Las aplicaciones híbridas potencializan las dos tecnologías para el desarrollo de aplicaciones móviles, ya que una de las principales desventajas de la tecnología web es que tiene un mayor consumo de recursos, producido por que se ejecuta dentro de un *WebView*, lo que produce que se renderize la aplicación e interprete como nativa, por lo que en algunos casos es necesario complementar con las tecnologías nativas para aumentar el rendimiento de la aplicación.

Personalmente, para un tamaño pequeño o mediano de una aplicación que sólo tiene que mostrar el contenido, que necesita utilizar componentes básicos del dispositivo y no se necesita un mayor rendimiento, se puede realizar mediante tecnología Web, sin embargo, si la aplicación se necesita un mayor consumo de recursos y el uso nativo de componentes se debe desarrollar con Android nativo[17].

2.4 Especificaciones de la tecnología de desarrollo.

Para el desarrollo de la aplicación en el marco del proyecto “Pequeñas y Pequeños Científicos” de la Universidad Politécnica Salesiana, se procederá a utilizar las tecnologías Android, HTML5, CSS3 y JavaScript, para la creación de componentes visuales y multimedia.

A continuación se detalla la instalación y configuración para la creación de aplicaciones híbridas en Windows 7 Ultimate.

2.4.1 Instalación y configuración de JDK 7

Para instalar el entorno de desarrollo JDK 7 en Windows, debemos seguir estos pasos:

1. Descargar el JDK de la sitio web de Oracle <http://www.oracle.com/technetwork/java/javase/downloads/index.html>, en esta página encontramos las diversas versiones (durante la redacción de estas líneas, la última versión que existe es la versión 7).

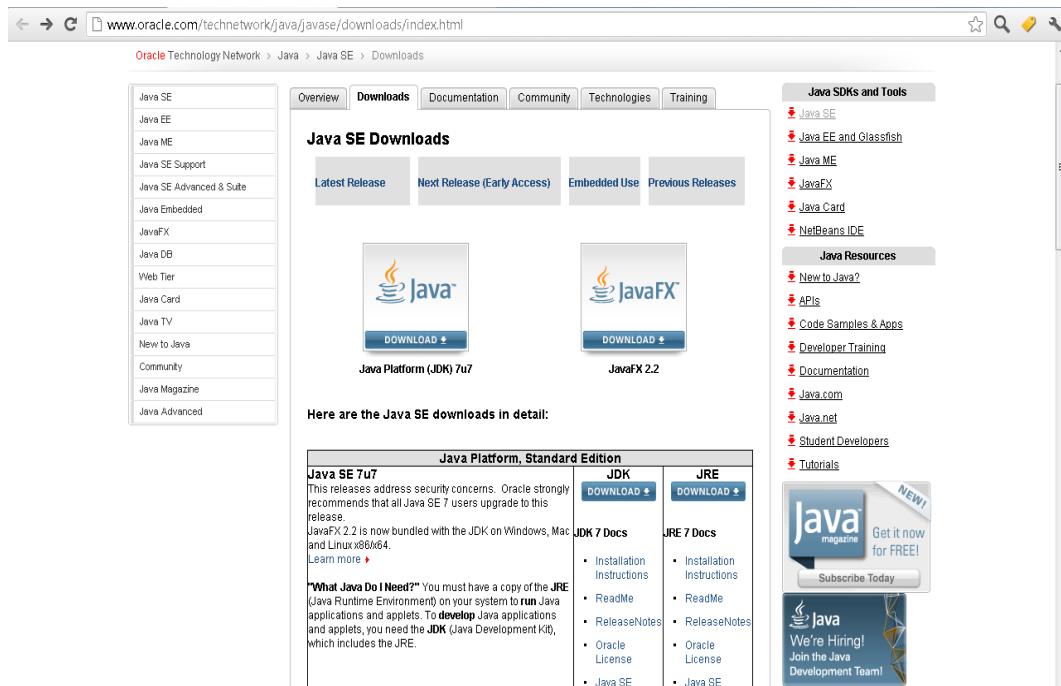


Ilustración 15 Sitio web de Oracle para la descarga del JDK

2. Se procede a aceptar los términos de licencia y escoger el instalador del sistema operativo en el que se está desarrollando, en este caso se escoge Windows.

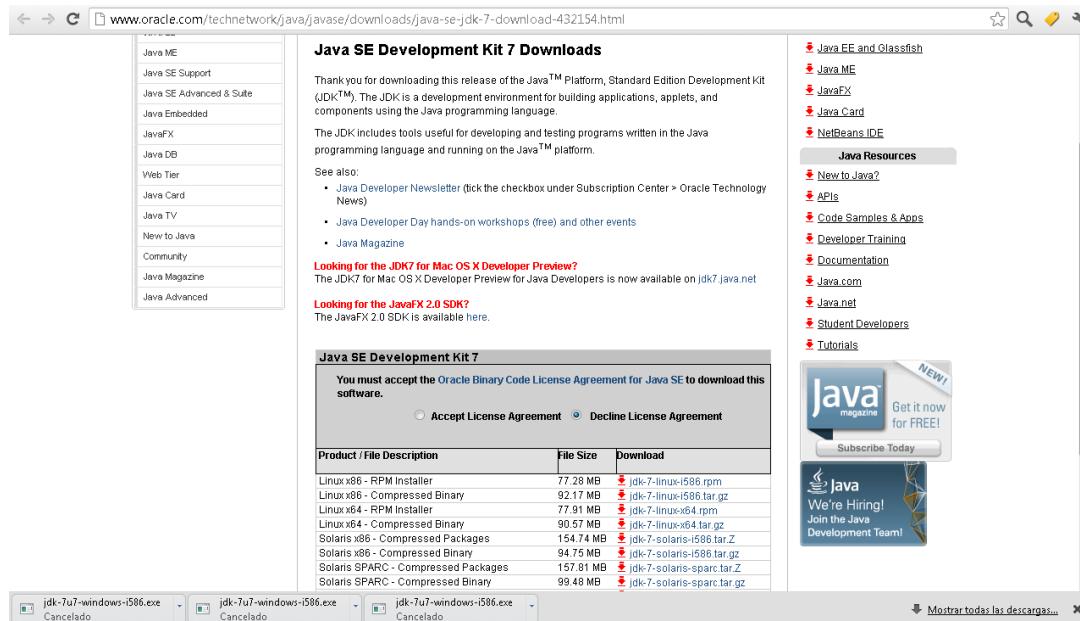


Ilustración 16 Descarga del JDK

3. Con esto se descargará un archivo ejecutable, en el cual procedemos a dar doble clic para la instalación y posteriormente damos clic en siguiente.



Ilustración 17 Instalación de JDK 7

4. Escogemos las características que deseamos instalar y damos clic en siguiente, se procederá a instalar del JDK.

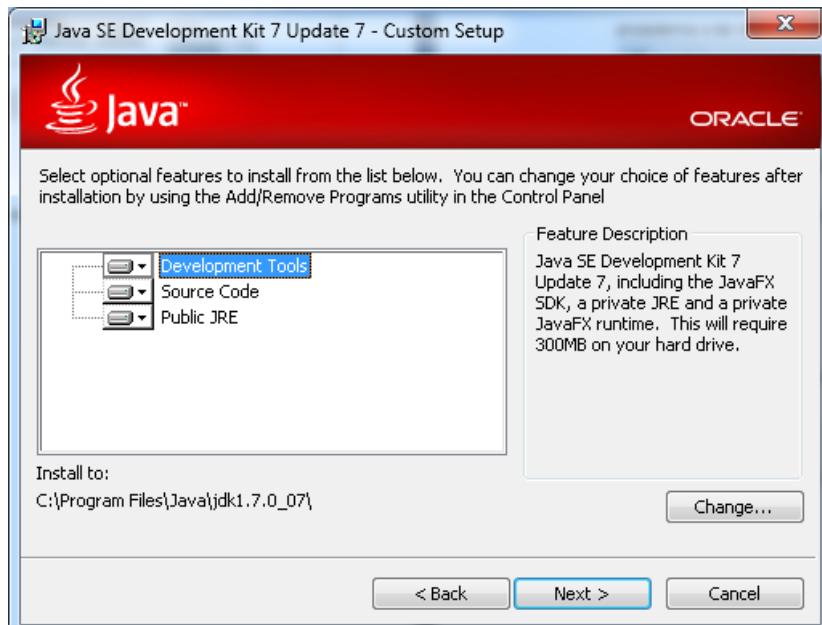


Ilustración 18 Características de instalación del JDK

5. Por último se nos presenta un mensaje que nos informa que la instalación se realizó con éxito, damos clic en cerrar.



Ilustración 19 Estado de la instalación del JDK

2.4.2 Instalación y configuración del SDK de Android

1. Primeramente se descargara la versión de SDK de Android del sitio web <http://developer.android.com/sdk/index.html>.

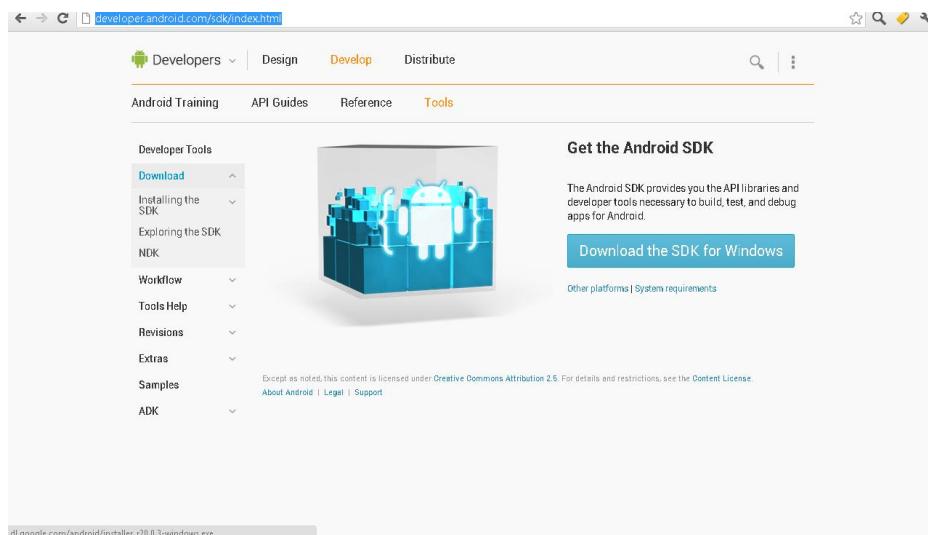


Ilustración 20 Sitio web para la descarga del SDK de Android

2. Se procede a la instalación del SDK, dando doble clic en el archivo descargado, se nos presenta la siguiente pantalla y damos clic en siguiente.

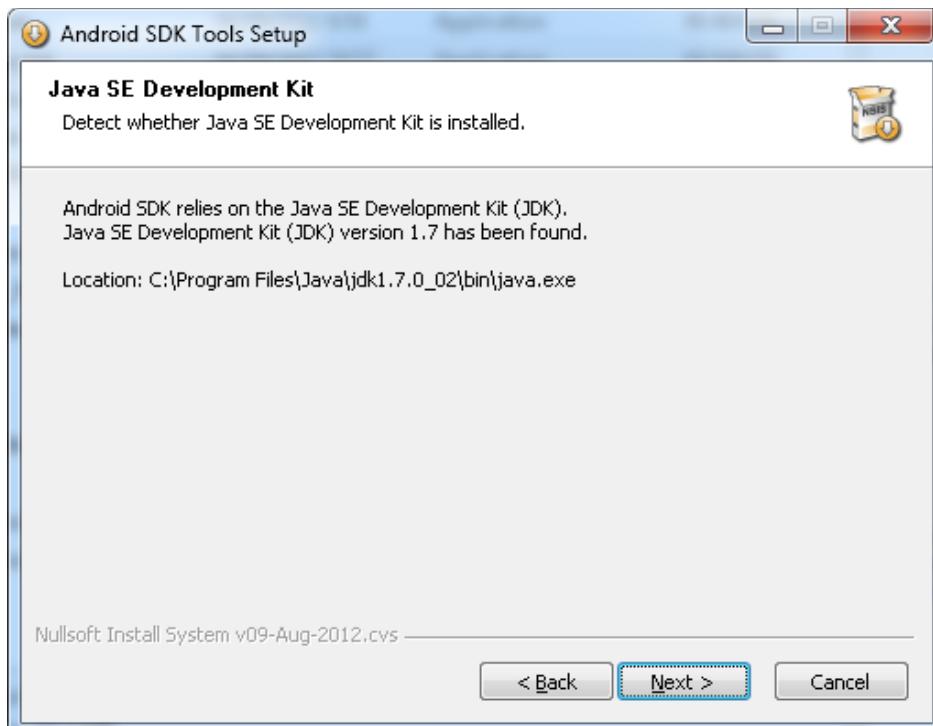


Ilustración 21 Instalación del SDK de Android

3. Escogemos el directorio en donde queremos que se instale el SDK y procedemos a dar clic en siguiente.

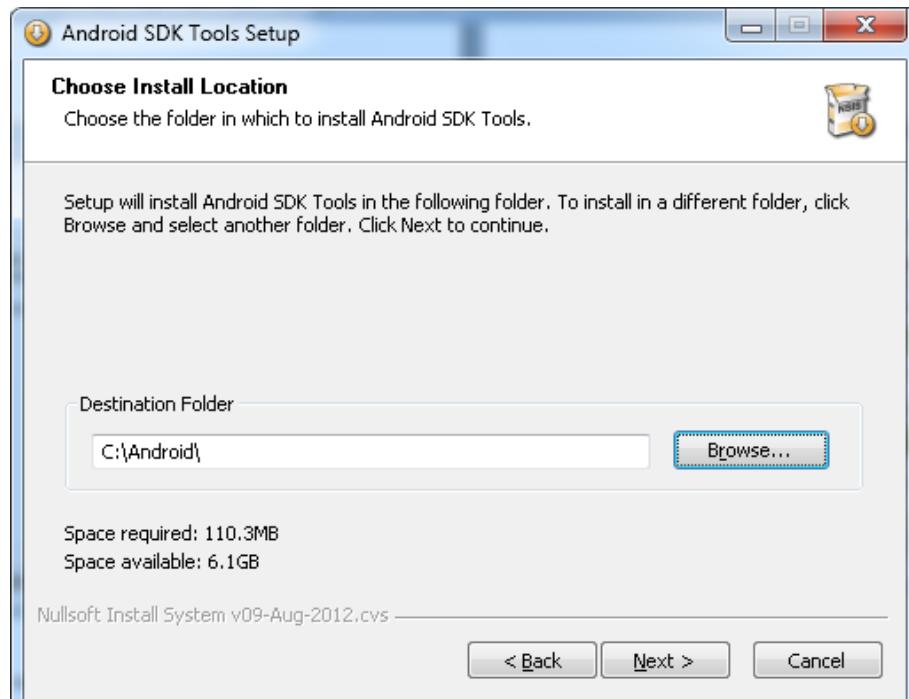


Ilustración 22 Directorio de instalación del SDK de Android

4. Se instalaran todos los paquetes del SDK de Android, damos clic en instalar.

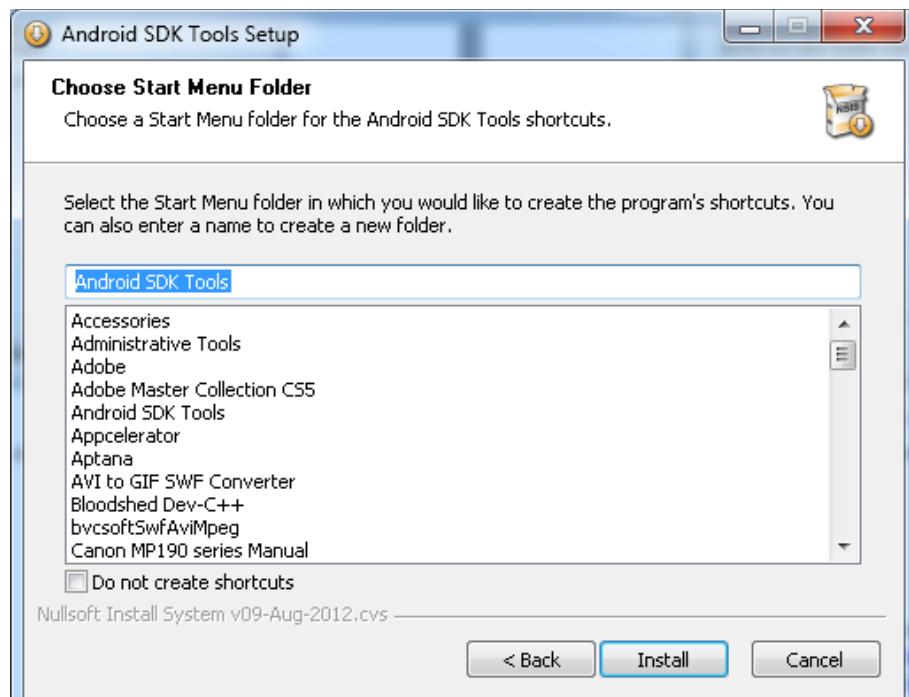


Ilustración 23 Paquetes de instalación de Android

5. Una vez completada la instalación procedemos a dar clic en Finalizar.
6. Posteriormente se deben instalar paquetes adicionales del sistema operativo Android, con esto podemos ejecutar nuestras aplicaciones dentro de un dispositivo virtual, escogemos la última versión de Android, la cual al momento de la redacción de este documento es la 4.1 (API 16) y se escogerá la versión 3.1 (API 11) como base, para realizar las pruebas en las diferentes versiones, damos clic en instalar paquetes.

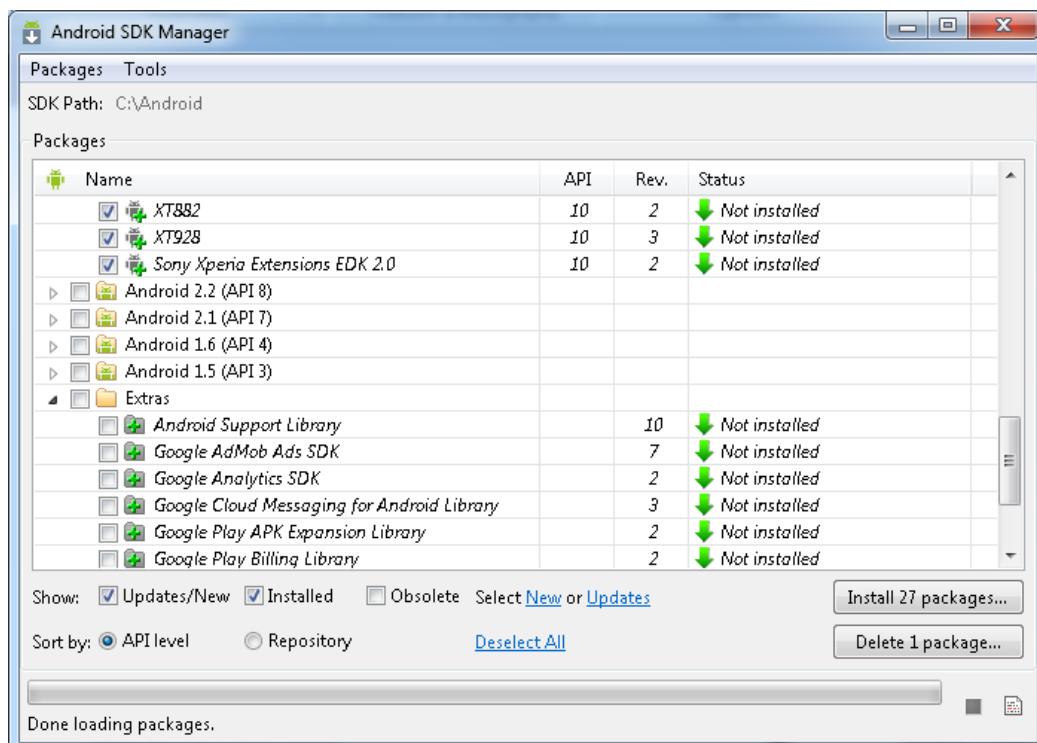


Ilustración 24 Instalación de las plataformas disponibles de Android

7. Una vez terminada la instalación de los paquetes se procederá a crear una máquina virtual de Android, para esto nos dirigimos al Menú *Tools* y seleccionamos *Manager ADVS*, aquí podemos crear los dispositivos virtuales para la ejecución de las aplicaciones en un entorno virtual.

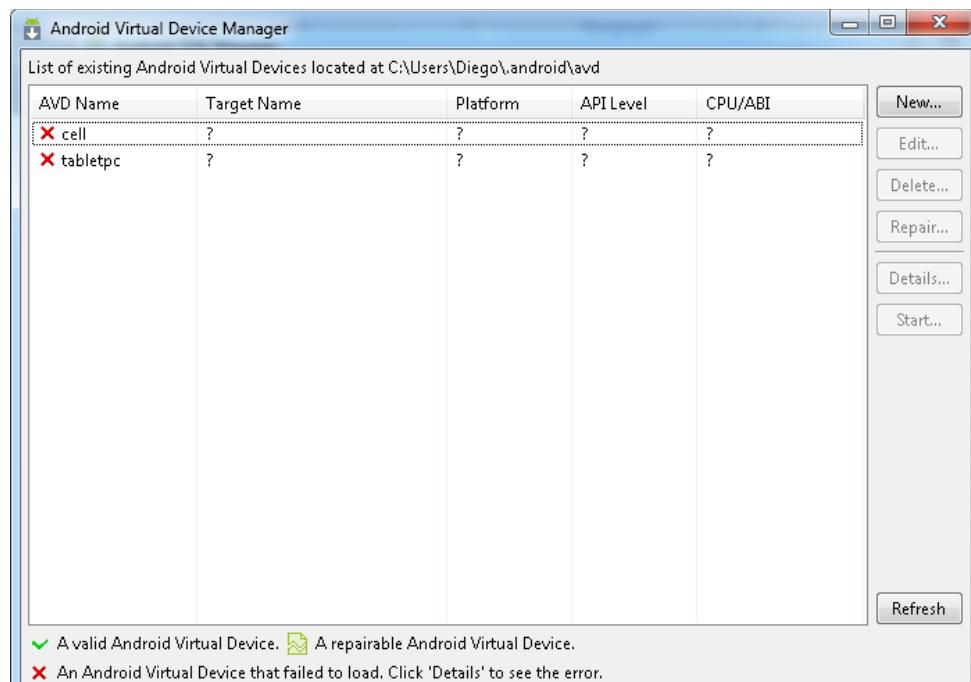


Ilustración 25 Administración de dispositivos virtuales de Android

8. Para crear un nuevo dispositivo virtual damos clic en Nuevo.

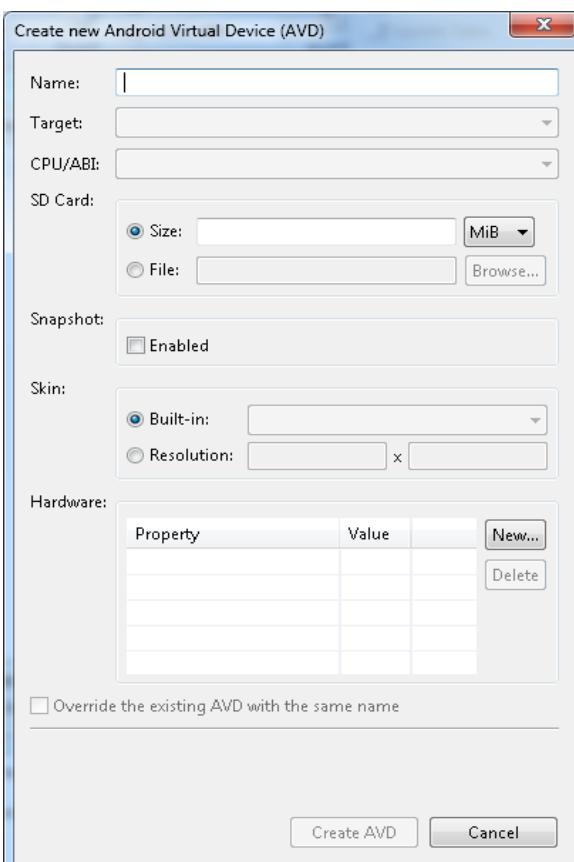


Ilustración 26 Creación de un nuevo dispositivo virtual de Android

9. Ingresamos el nombre del dispositivo virtual, la plataforma en la que se va a ejecutar, asignamos el tamaño de memoria, el tipo de pantalla y el hardware con el que cuenta el dispositivo. Una vez terminado damos clic en *Create ADV*.

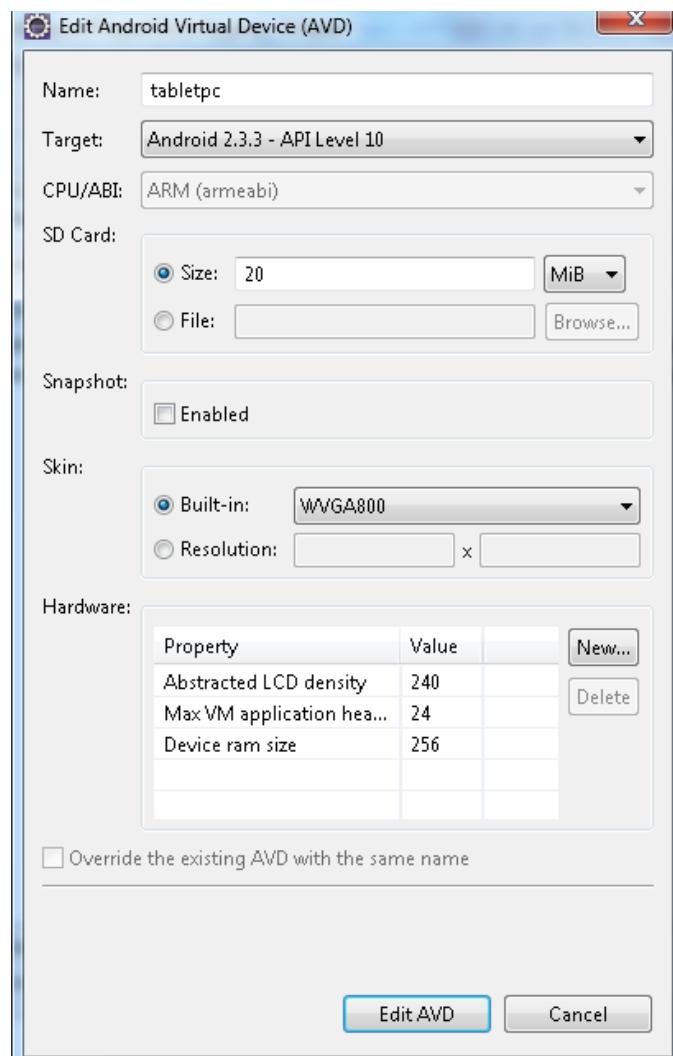


Ilustración 27 Configuración de un nuevo dispositivo virtual de Android

10. Una vez terminada la creación del dispositivo virtual, podemos iniciarla para comprobar su correcto funcionamiento, para esto damos clic en comenzar.

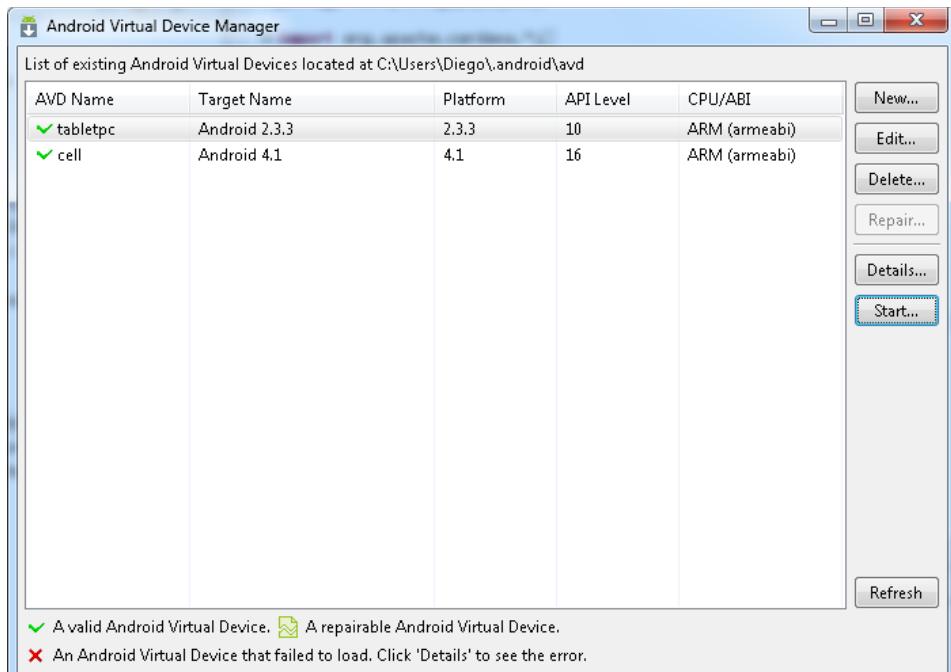


Ilustración 28 Visualización de los dispositivos virtuales Android.

11. En la siguiente pantalla se muestra la ejecución del dispositivo virtual.

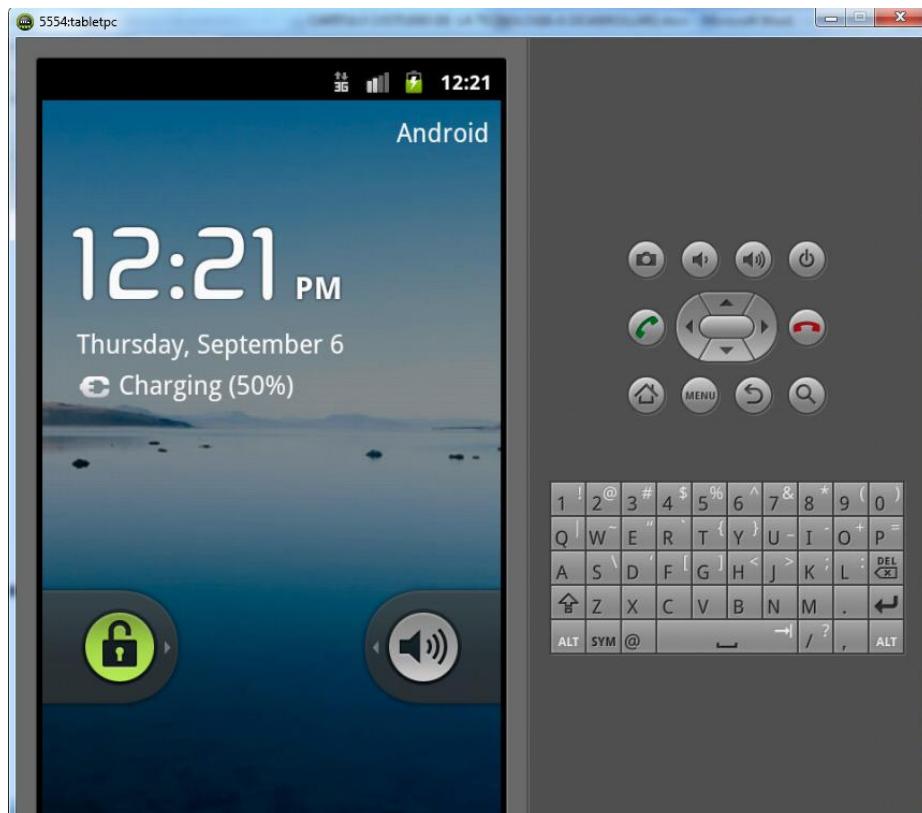


Ilustración 29 Ejecución del dispositivo virtual Android

2.4.3 Instalación y configuración de Eclipse Indigo

Se procederá a detallar la instalación y configuración del entorno de desarrollo eclipse

1. Se utilizará Eclipse Indigo 3.7 como entorno de desarrollo, el cual se descarga del sitio web de Eclipse: <http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/juno/R/eclipse-jee-juno-win32.zip>, dentro de este sitio escogemos el servidor de descarga.

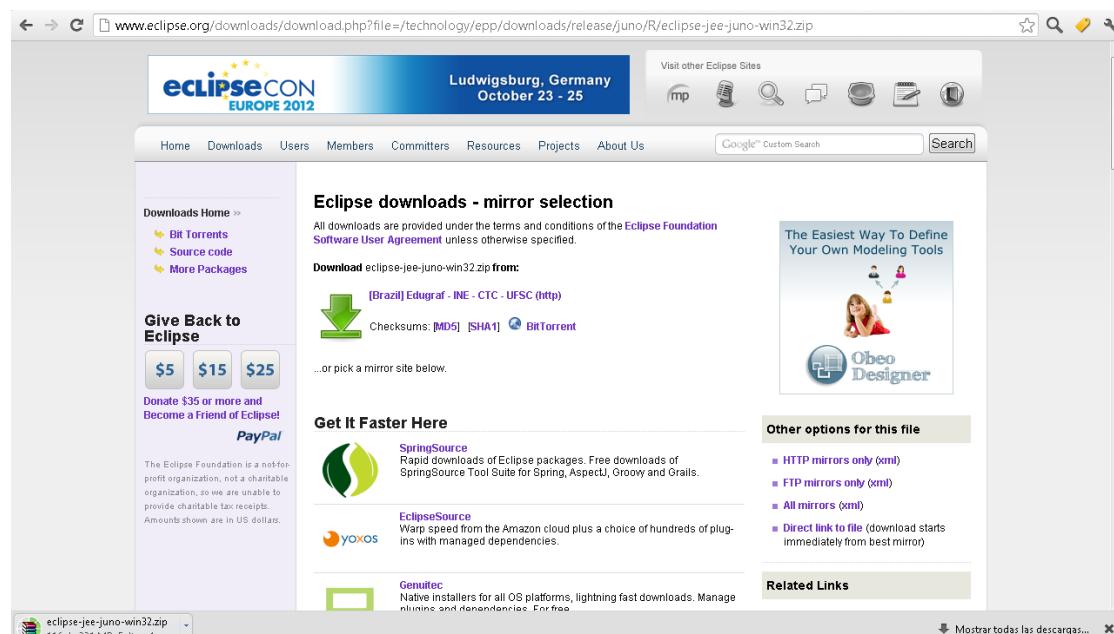


Ilustración 30 Sitio web de descarga del Eclipse Indigo 3.7

2. Una vez finalizada la descarga no se debe realizar ningún proceso de instalación, simplemente se debe descomprimir el fichero en un nuevo directorio y damos clic en el ejecutable de eclipse, la primera vez que se ejecuta la aplicación nos solicita escoger nuestro espacio de trabajo (*workspace*), carpeta donde se almacenarán todas las aplicaciones desarrolladas.

2.4.4 Instalar el plugin de Android para Eclipse

El siguiente paso es instalar el *plugin* específico de Android para la plataforma de desarrollo EclipseIndigo, para esto nos valemos de la herramienta ADT (*AndroidDevelopment Tools*) que nos facilita enormemente el desarrollo de proyectos basados en Android.

1. Seleccionamos en el Menú *Help>Install New Software*

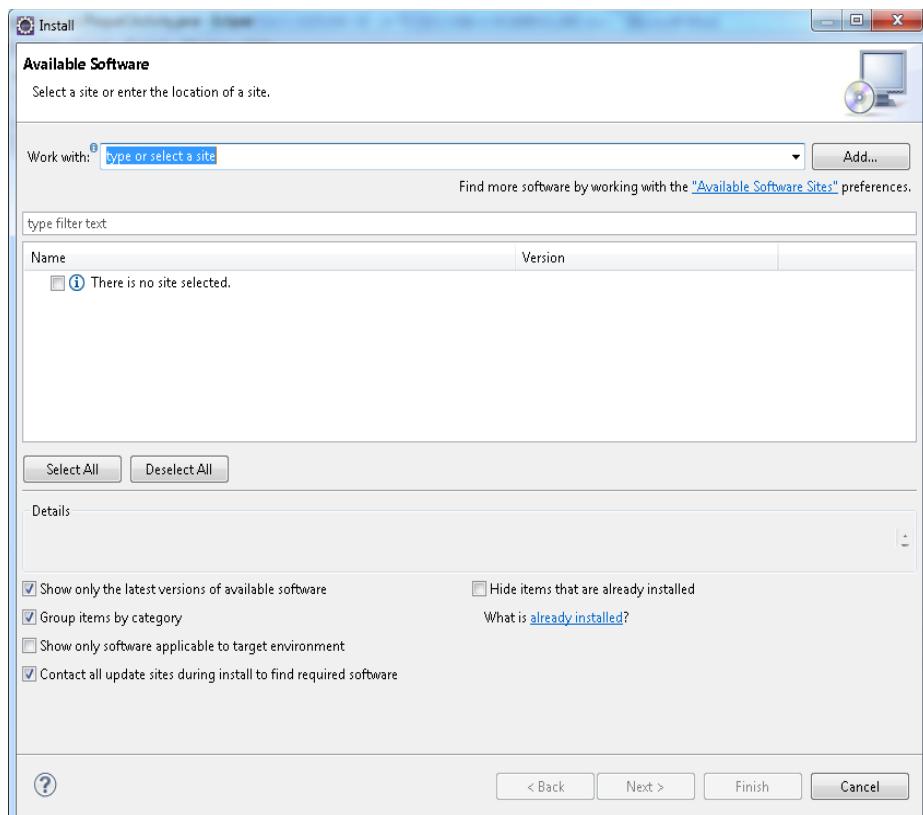


Ilustración 31 Instalación del plugin ADT en Eclipse.

2. Damos clic en agregar repositorio, le asignamos un nombre al repositorio que vamos a agregar (por ejemplo, Android) y en la localización ingresamos el siguiente repositorio web <https://dl-ssl.google.com/android/eclipse/> como se muestra en la ilustración 32.

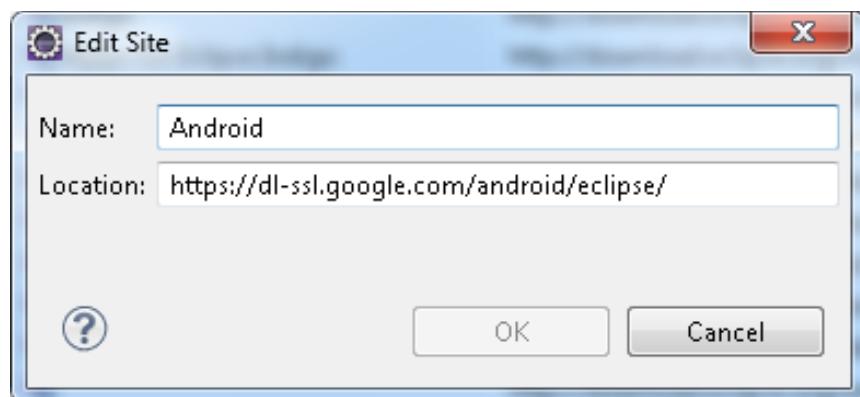


Ilustración 32 Agregación el repositorio de Android

3. Seleccionamos la pestaña *Development Tools* y procedemos a dar clic en siguiente, produciendo que se instalen los paquetes necesarios para el desarrollo de software en Android.

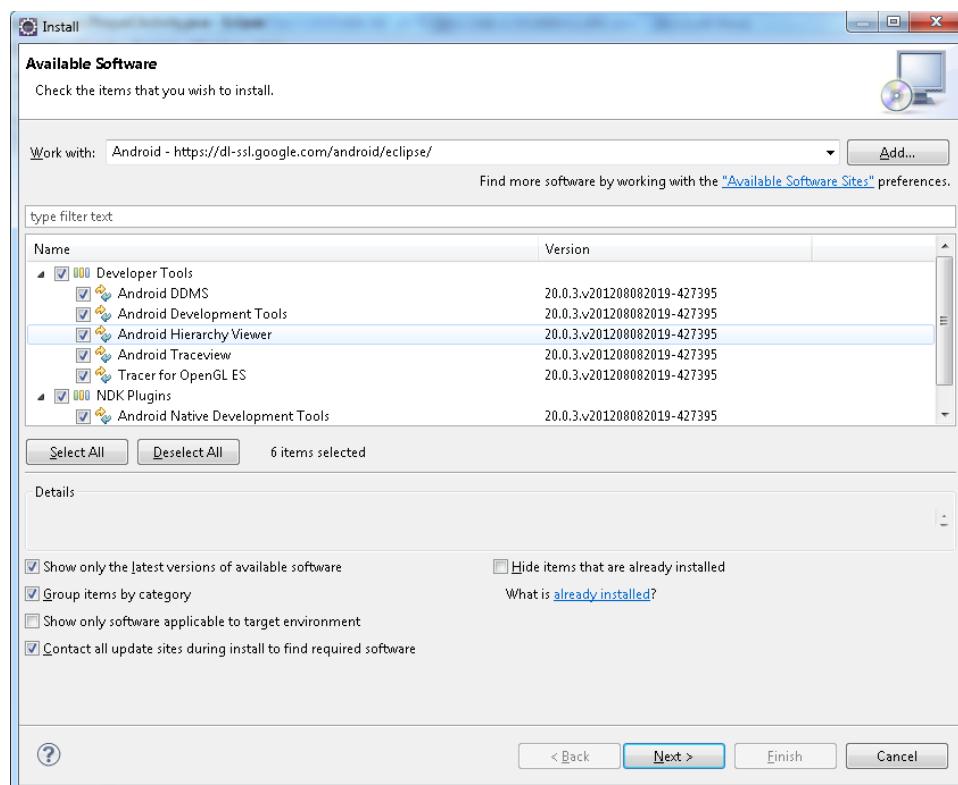


Ilustración 33 Selección e instalación de los paquetes de Android

4. Una vez terminada la instalación se deberá reiniciar Eclipse para que los cambios tengan efecto.

2.4.5 Creación de un nuevo proyecto Android

1. Seleccionar en el Menú *File > New > Project*, y en el menú resultante desplegar la opción *Android* y elegir *Android Project* y damos clic en siguiente.

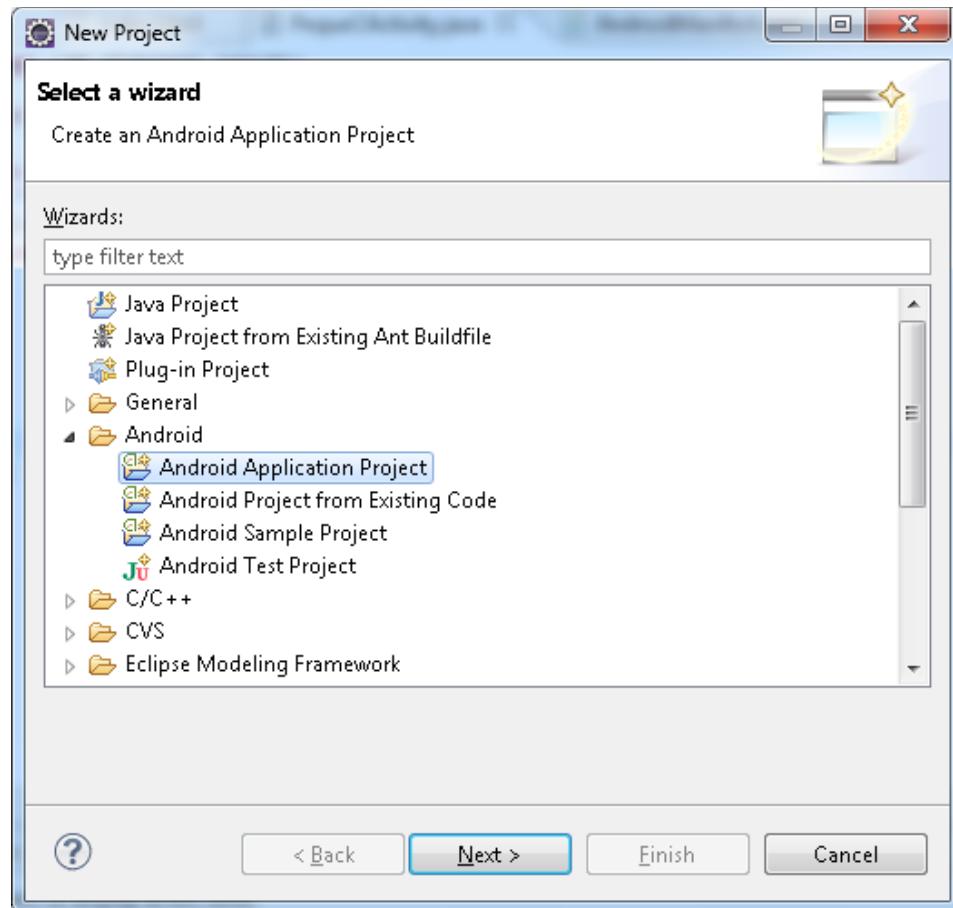


Ilustración 34 Creación de nuevo proyecto de Android

2. Se nos presenta la configuración del nuevo proyecto, el cual tiene los siguientes parámetros [14]:
 - *Project name*: Es el nombre del proyecto, en la práctica será el nombre que recibirá la carpeta donde se guardará todo lo relativo al presente proyecto dentro del *workspace*.
 - *Packagename*: El nombre del paquete bajo el cual será desarrollado todo el código.
 - *Activityname*: Es el nombre de la clase *Activity* principal que será creada, esta clase simplemente es una clase ejecutable, capaz de realizar alguna tarea y es imprescindible en la mayoría de las aplicaciones para Android.
 - *Applicationname*: El nombre de la aplicación que se va a desarrollar, constituye el nombre visible para el usuario del dispositivo móvil.

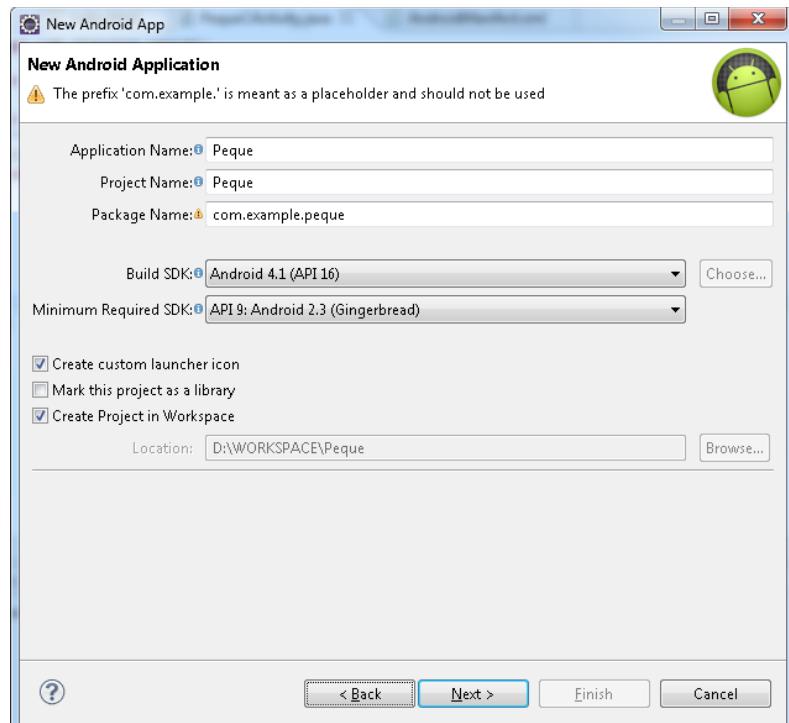


Ilustración 35 Configuración de un nuevo proyecto Android.

3. Una vez terminada la creación del nuevo proyecto, damos clic en finalizar.
4. Para ejecutar la aplicación debemos dar clic derecho sobre el proyecto y seleccionamos *Run As >AndroidApplication*, en la ilustración 36 se observa el resultado de ejecutar la aplicación.

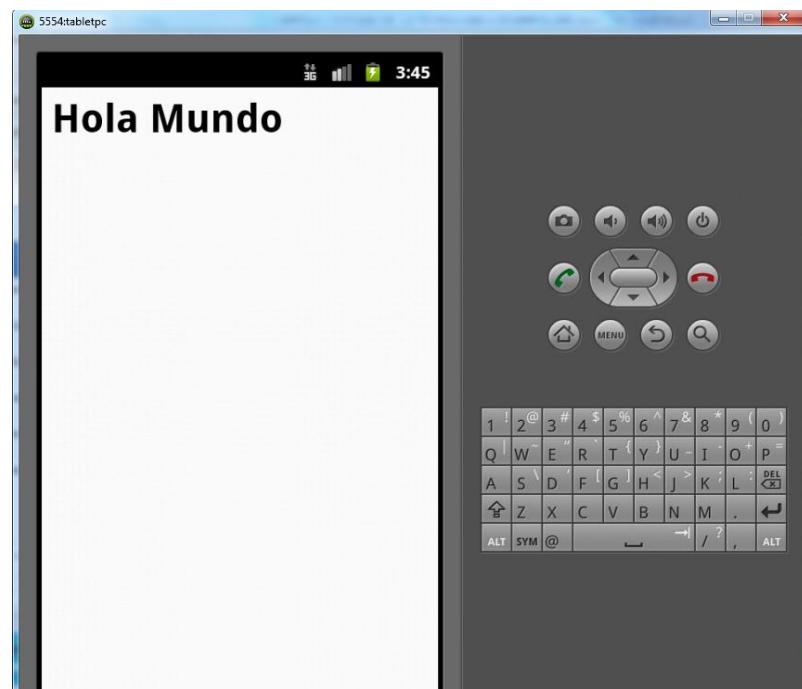


Ilustración 36 Ejecución de la aplicación Android.

CAPÍTULO 3

3 DISEÑO DEL SOFTWARE.

3.1 Levantamiento de Requerimientos de Software.

Como se mencionó en el capítulo anterior, el proyecto ya cuenta con un portal web (www.pequeciencia.org) en donde se encuentran los cuatro módulos desarrollados. Dichos módulos han sido desarrollados bajo la tecnología de Adobe Flash Builder. Un gran problema de la tecnología flash es que en las nuevas versiones de los sistemas operativos para dispositivos móviles como IOS y Android 4.1 se descartó la compatibilidad de ejecución de flash[33].

La tecnología está en auge y los sistemas operativos móviles están obteniendo una mayor compatibilidad con HTML5, CSS3 y JavaScript para plasmar animaciones que antes solo era posible en flash, consecuentemente, la principal fuente para el levantamiento de requerimientos es el sitio web y donde los códigos fuentes de la aplicación, repositorio de imágenes y sonidos fueron entregados por el equipo de desarrollo del proyecto de Pequeñas y Pequeños Científicos (Ing. Elena Duran y Ing. Diego Granja).

3.1.1 Definición de Stakeholders¹²

En la siguiente matriz se puede apreciar cuáles son los *stakeholders* involucrados en el proyecto de Pequeñas y Pequeños Científicos de la Universidad Politécnica Salesiana.

Matriz de los *Stakeholders*

NOMBRE	CARGO	RELEVACIÓN (1-5)
Ing. Elena Duran	Desarrolladores del Proyecto Pequeñas y Pequeños Científicos.	5
Ing. Diego Granja	Desarrolladores del proyecto Pequeñas y Pequeños Científicos.	5
Ing. Vladimir Robles	Director de Tesis	5
Ing. Rodolfo Bojorque	Equipo de Investigación	4
Ing. Eduardo Calle	Jefe del departamento de Investigación	3

Tabla 2 Matriz de los *Stakeholders*.

¹²Stakeholder es una persona u organización que tiene influencia directa, indirectamente o se ve influenciado por un proceso de desarrollo del software [25].

3.1.2 Técnicas de Elicitación [26]

El propósito de la elicitudes recolectar información de los *stakeholders* y capturar esa información en el formato de posibles requerimientos (tanto técnicos como no técnicos), junto con la justificación que da soporte a cada requerimiento, empleando técnicas como:

- Entrevistas: La entrevista es una forma de iteración simple entre el analista y los *stakeholders*. El analista simplemente permite que los *stakeholders* hablen sobre las tareas que debe cumplir el software, lo cual nos permite tener una visión global del dominio del problema. Por ello, se mantuvo entrevistas constantes obteniendo una visión completa del problema.
- Tormenta de Ideas: Se reúne un conjunto de personas relacionadas al producto y éstas sugieren y exploran sus ideas libremente, generando una amplia variedad de puntos de vistas, logrando construir una imagen más completa del problema. Se obtuvo numerosas alternativas para resolver el dominio del problema y las tecnologías a emplear.
- RAD (*Rapid ApplicationDesign*): Diseño Rápido de Aplicaciones, es un proceso de desarrollo de software que permite la construcción de sistemas útiles en menos de 90 días. Es una metodología para comprimir el análisis, el diseño, la construcción y las fases de prueba en una serie de ciclos de desarrollo cortos e iterativos, por lo que en cada iteración se entrega una parte funcional del sistema
- Prototipación: La prototipación es el proceso de creación de un sistema que ilustra las características relevantes del mismo. Comienza por el estudio preliminar de los requerimientos del usuario y concluye con una serie de requerimientos formulados y un prototipo desecharable. Es un proceso iterativo que permite la definición del comportamiento del sistema y en particular la apariencia del mismo.

3.1.3 Suposiciones y Dependencias

- Se brindará el apoyo del equipo de desarrolladores del proyecto Pequeñas y PequeñosCientíficos.
- La Universidad Politécnica Salesiana proveerá las licencias para el desarrollo e implementación del software.
- Acceso al laboratorio del CIDII (Centro de Investigación, Desarrollo e Innovación en Ingenierías de la UPS).
- La Universidad Politécnica Salesiana proveerá de 2 *tablets* con Android de 10 y 7 pulgadas.
- Conexión a internet.

3.1.4 Requerimientos Funcionales

La aplicación desarrollada consta del mundo eléctrico y del mundo verde, consecuentemente el Mundo Verde está compuesto de los siguientes personajes, temas y actividades.

Personajes:

Científica: Doctora Atómica.

Científico: Doctor Voltio

Niño: Panchito

Niña: Robertina

Árbol: Calipto

Temas:

Tema 1: Un gran problema

Subtema 1: La contaminación ambiental

Tramo 1: Planeta enfermo

Tramo 2: La contaminación

Subtema 2: Todo necesita energía

Tramo 1: Energía

Tramo 2: Polución

Tramo 3: Mundo verde

Tema 2: Las energías renovables

Subtema 1: Generación de energía eléctrica

Tramo 1: Energías renovables

Tramo 2: Energía eólica

Tramo 3:Aero generadores

Tramo 4:Energía hidráulica

Tramo 5:Energía solar

Tramo 6:Reto solar

Tema 3: Cuidando el medio ambiente

Subtema 1: Salvemos al planeta

Tramo 1: Ahorremos energía eléctrica

Tramo 2: Recomendaciones

Tramo 3: Más Recomendaciones

Actividad:

Experimento 1: Moviendo un motor con la energía del sol.

Materiales:

Un fuerte sol

1 panel solar fotovoltaico

2 cables con lagartos

Motorcito

Procedimiento:

1. Anda a un lugar en donde haya mucho sol.
2. Conecta los elementos como se indica a continuación

El Mundo Eléctrico consta con los siguientes personajes, temas y actividades. Lo cual se detallara a continuación.

Personajes

Científica: Doctora Atómica.

Científico: Doctor Voltio

Niño: Panchito

Niña: Robertina

Temas:

Tema 1: Un poquito de historia

Subtema 1: Un mundo lleno de energía

 Tramo 1: Todo necesita energía

Subtema 2: Tales de Mileto

 Tramo 1: Filósofo

Subtema 3: Benjamín Franklin y los rayos

 Tramo 1: Los rayos

 Tramo 2: Rayos = Electricidad

 Tramo 3: El pararrayos

Subtema 4: Alessandro Volta y la pila eléctrica

 Tramo 1: Pila eléctrica

Subtema 5: Thomas Alva Edison y el foco eléctrico

 Tramo 1: Se le prendió el “foco”

Tema 2: ¿Qué es la electricidad?

Subtema 1: Energía eléctrica

 Tramo 1: ¡Recuérdalo bien!

Subtema 2: El átomo y sus partes

 Tramo 1: El átomo

 Tramo 2: Electrones, protones y neutrones

Subtema 3: Tipos de electricidad

 Tramo 1: Electricidad estática

 Tramo 2: Corriente eléctrica

Tema 3: El circuito eléctrico

Subtema 1: La corriente eléctrica

 Tramo 1: El circuito eléctrico

 Tramo 2: El interruptor

 Tramo 3: Los conductores

 Tramo 4: Voltaje eléctrico

 Tramo 5: Batería eléctrica

 Tramo 6: Resistor eléctrico

Tema 4: Importancia de la electricidad

Subtema 1: La importancia de la electricidad

 Tramo 1: ¡Imagina...!

 Tramo 2: ¡Recuérdalo bien!

Actividades:

Experimento 1: Papelitos pegañosos.

Materiales:

- 1 electro-globo
- Pedazos de papel de colores
- Bitácora

Procedimiento:

1. Infla el electro-globo.
2. Esparce los papelitos de colores sobre tu mesa de trabajo.
3. Acerca el electro-globo que acabaste de inflar, y observa qué sucede. ¿Qué tal si hacemos lo que hizo Tales de Mileto?
4. Frota el electro-globo varias veces en tu ropa o en tu cabello.
5. Ahora sí, acerca el electro-globo a los papelitos de colores.

Experimento 2: Globos electrizados.

Materiales:

- 2 electro-globos
- Bitácora

Procedimiento:

1. Infla los dos electro-globos.
2. Acerca los globos a tu cabello. ¿Qué sucede?
3. Frota cada globo varias veces en tu cabello o en la chompa que tienes puesta. Anótalo en tu bitácora, y haz un dibujo de lo que observas.
4. Acerca los electro-globos a tu cabello, y pregúntale a tu compañero qué es lo que sucede con tu cabello.

Experimento 3: ¿Puedes prender un foco?

Materiales:

- 2 pilas y su portapilas
- 3 cables con "lagartos"
- 1 Foco con su boquilla
- 1 Interruptor
- Bitácora

Procedimiento:

Recuerda que la batería es necesaria para que exista la fuerza (voltaje) que mueve a los electrones. El foco será el resistor de nuestro primer circuito.

1. Conectaremos uno de los terminales del porta pilas con el interruptor usando un cable con lagartos.
2. Con otro cable con lagartos conecta el interruptor al foco.
3. Usando otro cable con lagartos conectaremos el otro terminal del foco hacia el otro terminal del portapilas.

3.1.5 Requerimientos no Funcionales

Rendimiento

- El sistema ofrecerá respuesta al usuario en tiempo real.
- El tiempo de respuesta promedio de la aplicación no debe superar los 5 segundos.

Seguridad

- Requisito de conexión.
- El sistema no contendrá ninguna conexión abierta de acceso a internet.

Fiabilidad

- Que el sistema sea 90% fiable, es decir, que la aplicación de cada 10 ejecuciones las 9 no tengan ningún error de compilación.

Disponibilidad

- En funcionamiento normal del sistema estará disponible el 100% del tiempo, esto se obtiene instalando la aplicación dentro del dispositivo móvil como una aplicación nativa.

Mantenibilidad

- El sistema estará en constante mantenimiento, consecuentemente se estarán actualizando o agregando nuevas funcionalidades.

Portabilidad

- Compatibilidad con plataformas: En el sistema desarrollado ofrece compatibilidad con otras plataformas Android desde la versión 3.0.
- Que se pueda descargar de cualquier parte del mundo gratuitamente por medio de *Google Play*.

Operabilidad

- Podrá ser operado por un niño de más de 8 años.

Interoperabilidad

- Interoperabilidad entre tecnología web y tecnología nativa Android.

3.1.6 Restricciones

El dispositivo móvil en donde se ejecutará la aplicación deberá tener los siguientes requisitos mínimos:

Procesador: ARM de 500Mhz
Memoria RAM: 512 MB.
Disco Duro: 30 MB disponibles.

Las versiones de sistema operativo de los dispositivos móviles deben ser:

Android Honeycomb 3.0 o superior

3.1.7 Evolución previsible del sistema

Se continuará el desarrollo de los siguientes módulos, los cuales están compuestos por los mundos eléctrico, magnético y digital.

Se implementarán nuevas funcionalidades a los módulos o mundos.

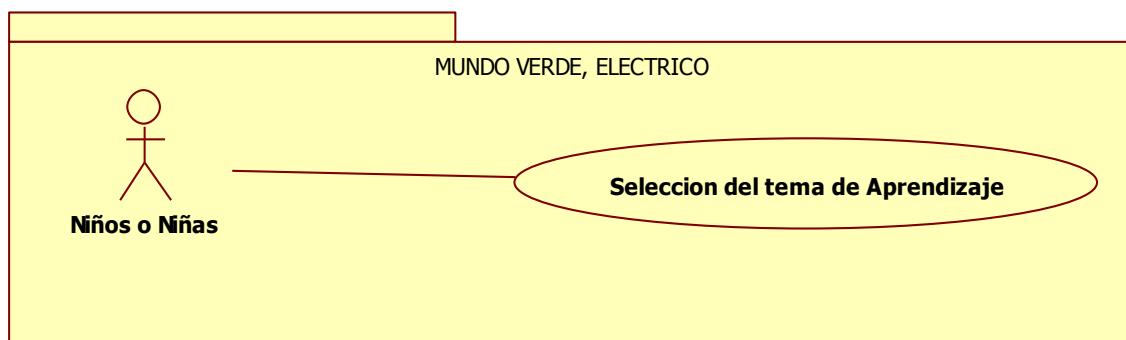
3.1.8 Validación de requerimientos

Los requerimientos serán validados por el equipo de desarrollo del proyecto de Pequeñas y Pequeños Científicos de la Universidad Politécnica Salesiana, el cual deberá cumplir con los requerimientos funcionales y no funcionales antes expuestos, elaborando un certificado de la validación que será firmado por la Ing. Elena Duran y el Ing. Diego Granja. Se adjunta en Anexo 1 el certificado de validación del proyecto.

3.2 Diseño de la aplicación.

3.2.1 Casos de Uso

Caso de uso: Aprendizaje Multimedia del Mundo Verde y Eléctrico

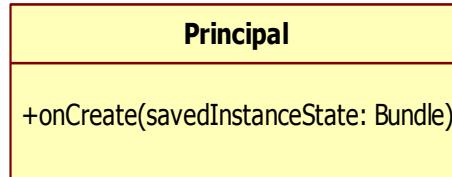


Descripción del caso de uso:

Paquete: Mundo Verde y Eléctrico	Iteración: 1	Minuta/s: 1.0		
Nombre Caso de Uso: Aprendizaje multimedia mundo verde y eléctrico.		Nro. Orden: R1		
Actor Principal: Niñas/os				
Prioridad: X Esencial	Útil	Deseable		
Complejidad: Simple Mediano Complejo		X Muy Complejo		
Tipo de Caso de Uso: XConcreto		Abstracto		
Objetivo: Enseñar el cuidado del medio ambiente desde los <i>tablets</i> con Android				
Precondiciones: Ingresar al sistema multimedia del mundo verde				
Pos condiciones:	Éxito: El usuario podrá aprender interactivamente el cuidado del medio ambiente y el eléctrico desde los <i>tablets</i> . Fracaso: Salida inesperada del sistema multimedia.			
Curso Normal:	Alternativas			
1. Visualización multimedia de los temas de aprendizaje del mundo verde y eléctrico.	Terminar cada uno de los temas de aprendizaje multimedia.			
Asociaciones de Extensión: No aplica				
Asociaciones de Inclusión: No aplica				
Caso de Uso donde se incluye: No aplica				
Caso de Uso donde se extiende: No aplica				
Case de Uso en donde se Generalización: No aplica				

3.2.2 Diagrama de Clases del Mundo Eléctrico y Verde

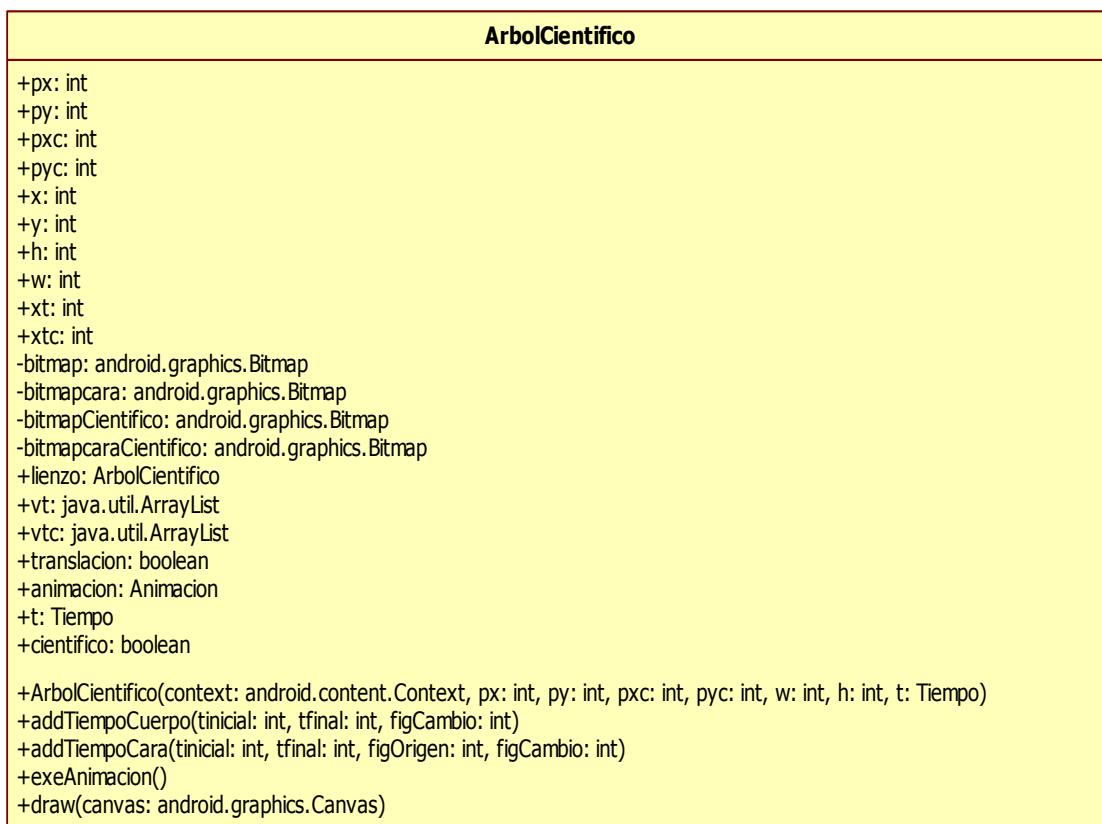
Diagrama de clases del Mundo Eléctrico.



La razón por la que el mundo eléctrico solo consta de una clase, es que dentro del método `onCreate()`, su única tarea es cargar un documento principal desarrollado en HTML5, CSS3 y JavaScript, el cual se encarga de todo el manejo de la aplicación.

Diagrama de clases del Mundo Verde.

En el mundo verde se desarrollo una aplicación híbrida del Mundo Verde. Dentro del marco el desarrollo nativo está compuesto por las siguientes clases:



NinoCientifica
<pre>+px: int +py: int +pxc: int +pyc: int +x: int +y: int +h: int +w: int +xt: int +xtc: int -bitmap: android.graphics.Bitmap -bitmapcara: android.graphics.Bitmap -bitmapCientifica: android.graphics.Bitmap -bitmapcaraCientifica: android.graphics.Bitmap +lienzo: NinoCientifica +vt: java.util.ArrayList +vtc: java.util.ArrayList +translacion: boolean +animacion: Animacion +t: Tiempo +cientifica: boolean +NinoCientifica(context: android.content.Context, px: int, py: int, pxc: int, pyc: int, w: int, h: int, t: Tiempo) +addTiempoCuerpo(tinicial: int, tfinal: int, figCambio: int) +addTiempoCara(tinicial: int, tfinal: int, figOrigen: int, figCambio: int) +exeAnimacion() +draw(canvas: android.graphics.Canvas)</pre>

Nina
<pre>+px: int +py: int +pxc: int +pyc: int +x: int +y: int +h: int +w: int +xt: int +xtc: int -bitmap: android.graphics.Bitmap -bitmapcara: android.graphics.Bitmap +lienzo: Nina +vt: java.util.ArrayList +vtc: java.util.ArrayList +translacion: boolean +animacion: Animacion +t: Tiempo +Nina(context: android.content.Context, px: int, py: int, pxc: int, pyc: int, w: int, h: int, t: Tiempo) +addTiempoCuerpo(tinicial: int, tfinal: int, figCambio: int) +addTiempoCara(tinicial: int, tfinal: int, figOrigen: int, figCambio: int) +exeAnimacion() +draw(canvas: android.graphics.Canvas)</pre>

Fondo

```
-bitmap: android.graphics.Bitmap  
-bitmapPantalla: android.graphics.Bitmap  
+fondo: Fondo  
+xt: int  
+xtp: int  
+h: int  
+w: int  
+px: int  
+py: int  
+x: int  
+y: int  
+pxp: int  
+pyp: int  
+t: Tiempo  
+animacion: Animacion  
+bajar: boolean  
+agrandar: boolean  
  
+Fondo(context: android.content.Context, px: int, py: int, ppx: int, pyp: int, w: int, h: int, t: Tiempo)  
+draw(canvas: android.graphics.Canvas)  
+exeAnimacion()
```

Web

```
+p: Principal  
+web: android.webkit.WebView  
+prbload: android.widget.ProgressBar  
+w: int  
+h: int  
+px: int  
+py: int  
  
+Web(p: Principal, px: int, py: int, w: int, h: int)  
+getButton(btn: android.widget.ImageButton, px: int, py: int, id: int): android.widget.ImageButton
```

Tiempo

```
+tiempo: int  
  
+Tiempo()  
+setTiempo(tiempo: int)  
+getTiempo(): int
```

Animacion

```
#doInBackground(params: String): boolean  
#onProgressUpdate(x: String)
```

Media

```
+media: android.media.MediaPlayer  
+Media(context: android.content.Context, audio: int)
```

Menu

```
-btnProblema: android.widget.ImageButton  
-btnRenovable: android.widget.ImageButton  
-btnCuidando: android.widget.ImageButton  
-btnActividad: android.widget.ImageButton  
+fondo: Fondo  
+media: Media  
+t: Tiempo  
+p: Principal  
+Menu(p: Principal, rl: android.widget.RelativeLayout, fondo: Fondo, web: Web, arbolCientifico: ArbolCientifico, ninoCientifica: NinoCientifica, nina: Nina, t: Tiempo, media: Media)  
+setEventos()  
+hilo()
```

Menu1

```
-btn1: android.widget.ImageButton  
-btn2: android.widget.ImageButton  
-btnMundo: android.widget.ImageButton  
+fondo: Fondo  
+web: Web  
+arbolCientifico: ArbolCientifico  
+media: Media  
+t: Tiempo  
+p: Principal  
+Menu1(p: Principal, rl: android.widget.RelativeLayout, fondo: Fondo, web: Web, arbolCientifico: ArbolCientifico, ninoCientifica: NinoCientifica, nina: Nina, t: Tiempo, media: Media)  
+setEventos()  
+hilo()  
+setTiempos()
```

Menu2

```
-btn1: android.widget.ImageButton  
-btnMundo: android.widget.ImageButton  
+fondo: Fondo  
+web: Web  
+arbolCientifico: ArbolCientifico  
+media: Media  
+t: Tiempo  
+p: Principal  
+Menu2(p: Principal, rl: android.widget.RelativeLayout, fondo: Fondo, web: Web, arbolCientifico: ArbolCientifico, ninoCientifica: NinoCientifica, nina: Nina, t: Tiempo, media: Media)  
+setEventos()  
+hilo()
```

Menu3

```
-btn1: android.widget.ImageButton  
-btnMundo: android.widget.ImageButton  
+fondo: Fondo  
+web: Web  
+media: Media  
+t: Tiempo  
+p: Principal  
+Menu3(p: Principal, rl: android.widget.RelativeLayout, fondo: Fondo, web: Web, arbolCientifico: ArbolCientifico, ninoCientifica: NinoCientifica, nina: Nina, t: Tiempo, media: Media)  
+setEventos()  
+hilo()
```

Menu4

```
-btn1: android.widget.ImageButton  
-btnMundo: android.widget.ImageButton  
+fondo: Fondo  
+web: Web  
+media: Media  
+t: Tiempo  
+p: Principal  
+Menu4(p: Principal, rl: android.widget.RelativeLayout, fondo: Fondo, web: Web, arbolCientifico: ArbolCientifico, ninoCientifica: NinoCientifica, nina: Nina, t: Tiempo, media: Media)  
+setEventos()  
+hilo()
```

Problema
<pre>-btnA: android.widget.ImageButton -btnB: android.widget.ImageButton -btnAtras: android.widget.ImageButton +fondo: Fondo +web: Web +arbolCientifico: ArbolCientifico -nina: Nina -ninoCientifica: NinoCientifica +media: Media +t: Tiempo +p: Principal +Problema(p: Principal, rl: android.widget.RelativeLayout, fondo: Fondo, web: Web, arbolCientifico: ArbolCientifico, ninoCientifica: NinoCientifica, nina: Nina, t: Tiempo, media: Media) +setEventos() +hilo() +setTiempos()</pre>

Necesita
<pre>-btnA: android.widget.ImageButton -btnB: android.widget.ImageButton -btnC: android.widget.ImageButton -btnAtras: android.widget.ImageButton +fondo: Fondo +web: Web +arbolCientifico: ArbolCientifico -nina: Nina -ninoCientifica: NinoCientifica +media: Media +t: Tiempo +p: Principal +Necesita(p: Principal, rl: android.widget.RelativeLayout, fondo: Fondo, web: Web, arbolCientifico: ArbolCientifico, ninoCientifica: NinoCientifica, nina: Nina, t: Tiempo, media: Media) +setEventos() +hilo() +setTiempos()</pre>

Renovables
<pre>-btnA: android.widget.ImageButton -btnB: android.widget.ImageButton -btnC: android.widget.ImageButton -btnD: android.widget.ImageButton -btnE: android.widget.ImageButton -btnF: android.widget.ImageButton -btnAtras: android.widget.ImageButton +fondo: Fondo +web: Web +arbolCientifico: ArbolCientifico -nina: Nina -ninoCientifica: NinoCientifica +media: Media +t: Tiempo +p: Principal +actividad: Actividad +Renovables(p: Principal, rl: android.widget.RelativeLayout, fondo: Fondo, web: Web, arbolCientifico: ArbolCientifico, ninoCientifica: NinoCientifica, nina: Nina, t: Tiempo, media: Media) +setEventos() +hilo() +setTiempos() +actividad()</pre>

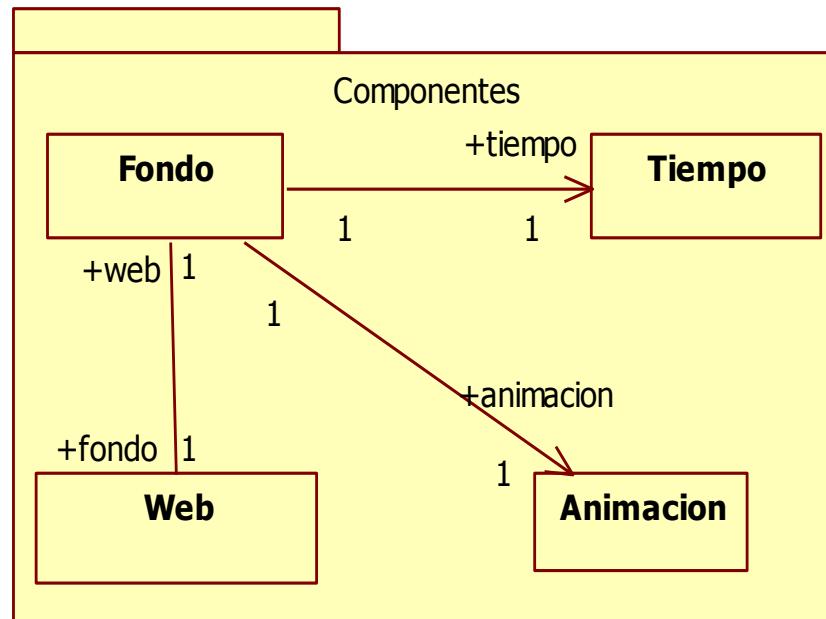
Cuidando
<pre>-btnA: android.widget.ImageButton -btnB: android.widget.ImageButton -btnC: android.widget.ImageButton -btnAtras: android.widget.ImageButton +fondo: Fondo +web: Web +arbolCientifico: ArbolCientifico -nina: Nina -ninoCientifica: NinoCientifica +media: Media +t: Tiempo +p: Principal +Cuidando(p: Principal, rl: android.widget.RelativeLayout, fondo: Fondo, web: Web, arbolCientifico: ArbolCientifico, ninoCientifica: NinoCientifica, nina: Nina, t: Tiempo, media: Media) +setEventos() +hilo() +setTiempos()</pre>

Actividad
<pre> -btn1: android.widget.ImageButton -btn2: android.widget.ImageButton -btn3: android.widget.ImageButton -btn4: android.widget.ImageButton -btn5: android.widget.ImageButton -btnAtras: android.widget.ImageButton +fondo: Fondo +web: Web -ninoCientifica: NinoCientifica +media: Media +t: Tiempo +p: Principal +tt: int +renovable: boolean +Actividad(p: Principal, rl: android.widget.RelativeLayout, fondo: Fondo, web: Web, arbolCientifico: ArbolCientifico, ninoCientifica: NinoCientifica, nina: Nina, t: Tiempo, media: Media, renovable: boolean) +setInvisible() +setVisible() +setEventos() +hid() +setTiempos() </pre>

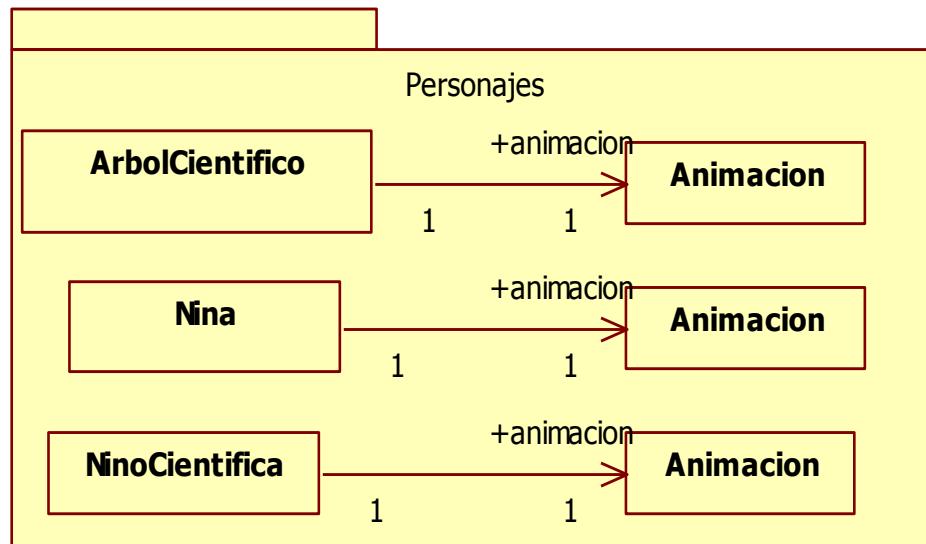
Principal
<pre> +fondo: Fondo +web: Web +arbolCientifico: ArbolCientifico -ninoCientifica: NinoCientifica -nina: Nina +t: Tiempo +estado: boolean +media: Media +rl: android.widget.RelativeLayout +btn: android.widget.ImageButton = null +escena: int +menu: Menu +menu1: Menu1 +menu2: Menu2 +menu3: Menu3 +menu4: Menu4 +problema: Problema +necesita: Necesita +renovables: Renovables +cuidando: Cuidando +actividad: Actividad +animacion: Animacion +onCreate(savedInstanceState: Bundle) +resetPersonajes() +setEscena(escena: int) #onResume() #onPause() </pre>

Las clases están empaquetadas dentro de los siguientes paquetes:

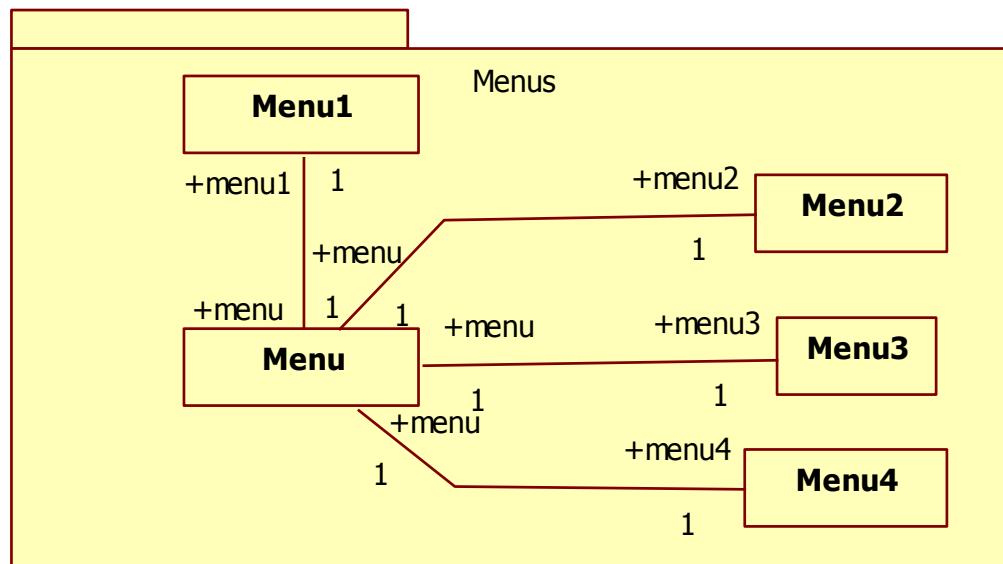
- Componentes:



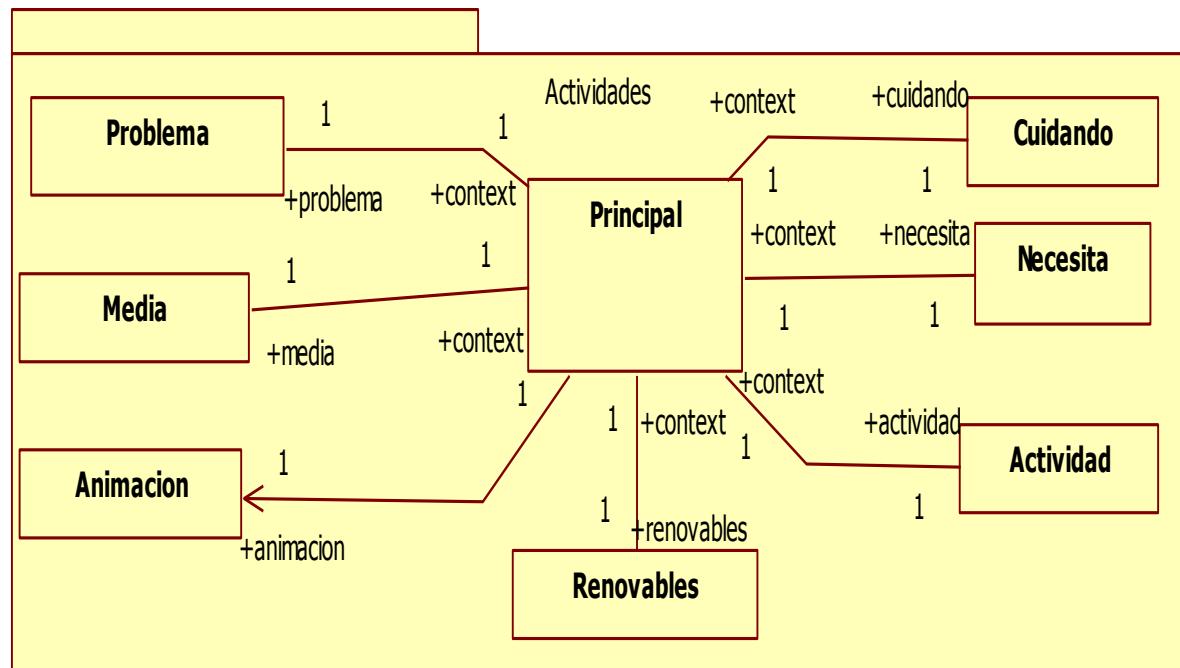
- Personajes



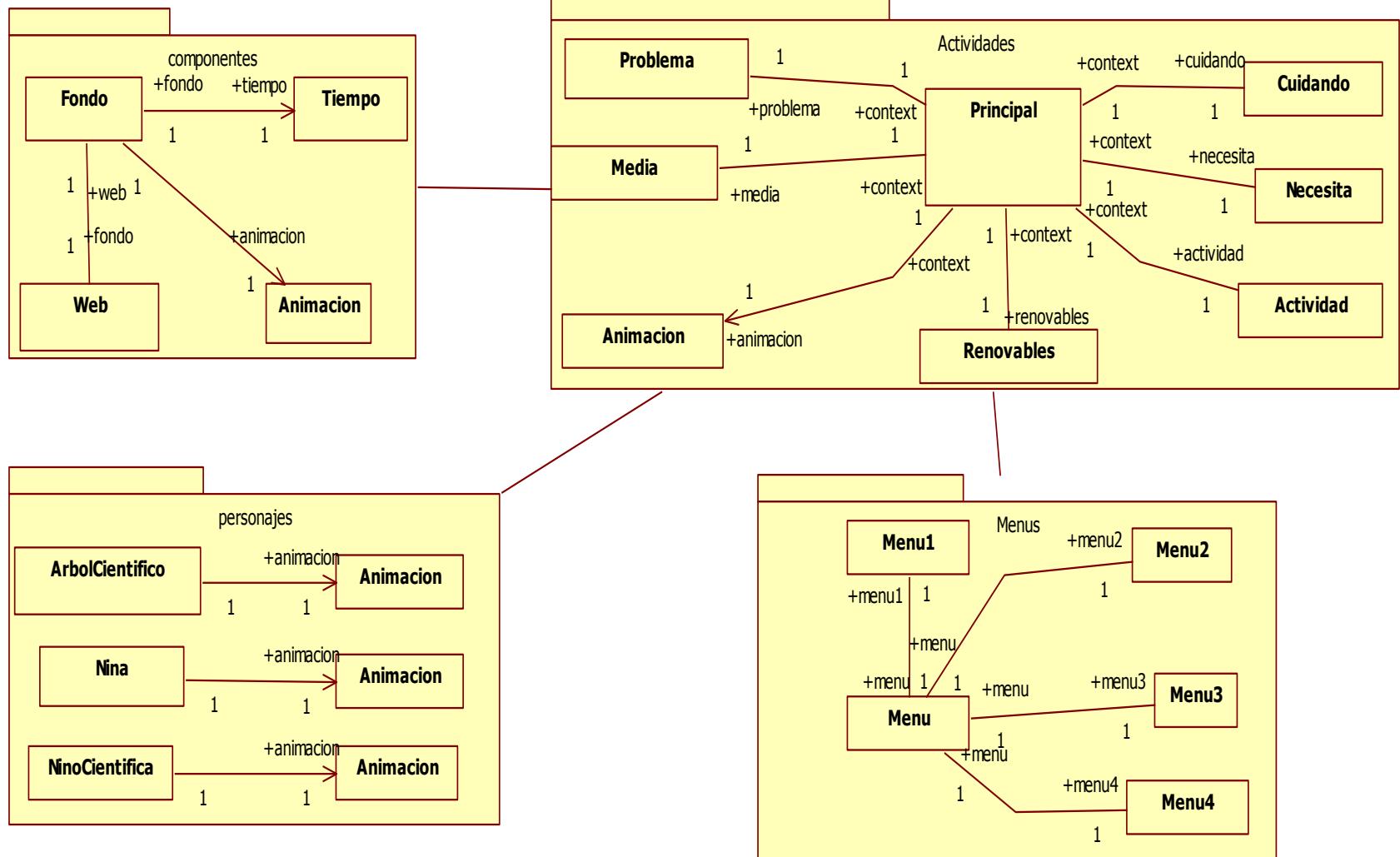
- Menús



- Actividades

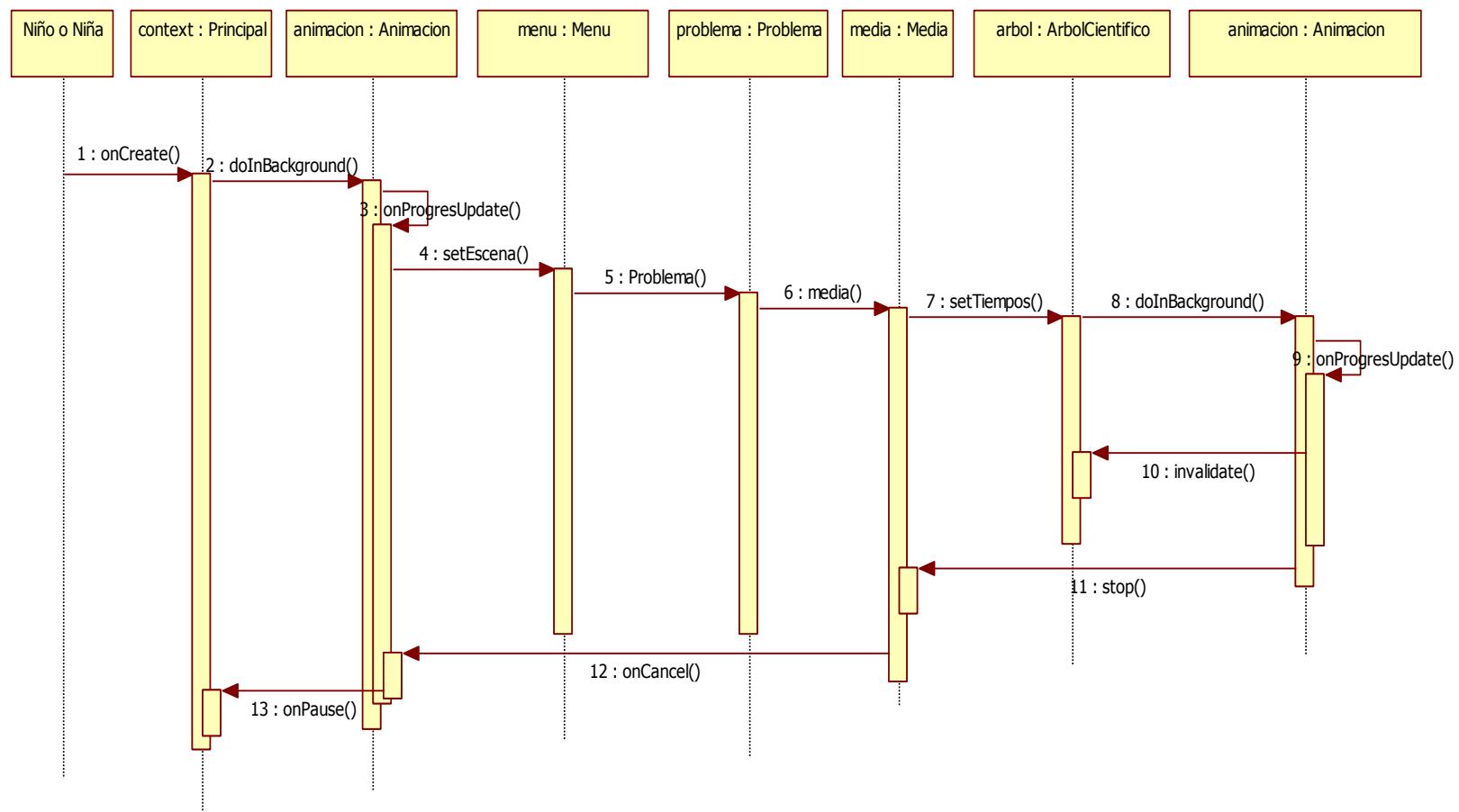


Finalmente se presenta el Diagrama UML del Mundo Verde.

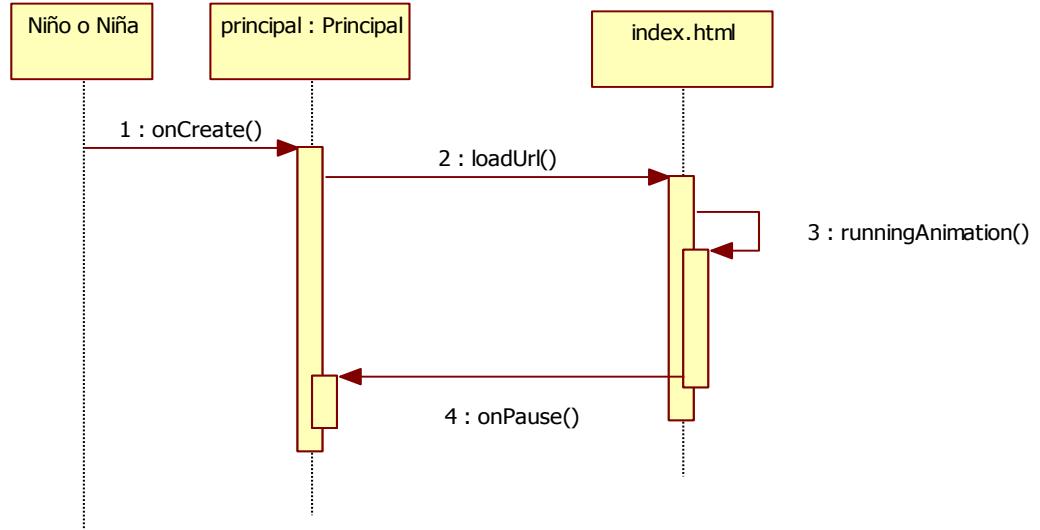


3.2.3 Diagramas de Secuencia del Mundo Eléctrico y Verde

Mundo Verde



Mundo Eléctrico



3.3 Análisis de las herramientas de desarrollo.

Como se mencionó en los capítulos anteriores se desarrollará una aplicación híbrida para el mundo verde, la cual constará del desarrollo en lenguaje nativo para la reproducción del sonido, animación de los personajes y del fondo de la aplicación, por lo que se emplearán los siguientes paquetes de Android:

3.3.1 Gráficos en Android

En el sistema Android se incluye la clase *Canvas* que implementa numerosos métodos para dibujar gráficos en pantalla. La manera más sencilla de dibujar gráficos en Android es dibujarlos sobre un *Canvas* asociado a un *View*[28].

Cada objeto *View* de una aplicación tiene asociado un objeto *Canvas*. Este objeto es pasado como argumento en las llamadas *onDraw* que realiza el sistema cuando es necesario presentar el *View*. Por lo tanto, para dibujar en el *Canvas* basta con sobrescribir (*Override*) el método *onDraw* [28].

Por ejemplo:

```
class Lienzo extends View {
    public Lienzo(Context context) {
        super(context);
    }
    @Override
    public void onDraw(Canvas canvas) {
        // Código para graficar dentro del Lienzo
    }
}
```

Para utilizar un recurso multimedia, basta con añadir el archivo al directorio `/res/drawable/` del proyecto, al hacer esto, el *plugin* de eclipse se encargará de generar automáticamente los identificadores java en la clase R. Con esto se puede hacer referencia desde el código o desde el diseño XML, se obtiene mediante un identificador de recursos (id) que es el nombre del archivo sin la extensión del tipo de archivo. Android soporta los formatos PNG (aconsejables), JPG (aceptables) y GIF (desaconsejados) [29].

Para dibujar un mapa de bits, tenemos que utilizar la clase *BitmapFactory*, lo cual enviamos como parámetro el identificador de referencia de la imagen dentro del método *decodeResource()* y nos retorna un objeto de tipo *Bitmap*.

Por ejemplo:

```
Bitmap bitmap = BitmapFactory.decodeResource(getResources(),  
R.drawable.imagen);
```

Para graficar el mapa de bits (*bitmap*), nos dirigimos al método *onDraw()* de la clase Lienzo, dentro del método *onDraw()* recibimos como parámetro un objeto de tipo *Canvas*, la cual tiene el método *drawBitmap()* que recibe los siguientes parámetros [30]:

- *Bitmap bitmap*: Es el mapa de bits.
- *Rect src*: Puede ser nulo, es el rectángulo del mapa de bits que se recortará del origen.
- *Rect dst*: Es el rectángulo que el mapa de bits que se mostrarán el *View*.
- *Paint paint*: Puedes ser nulo. El estilo que se usa para dibujar el mapa de bits.

3.3.2 *AsyncTask*[27]

Todos los componentes de una aplicación Android, tanto las actividades como los servicios, se ejecutan en el mismo hilo, llamado hilo principal (*mainthread*). Existen ocasiones en las que necesitamos que se realicen tareas en segundo plano, para solventar esto normalmente se hace uso de los hilos (*Thread*), el problema con los hilos es que la interfaz gráfica (UI) de Android no permite llamadas desde otros hilos que no sea el suyo, para resolver esto debemos crear un *AsyncTask* (Tarea Asíncrona).

La clase *AsyncTask* dispone de tres tipos de parámetros distintos, que deberemos especificar cuándo declaremos nuestra clase.

- *Params*: Datos que pasaremos al comenzar la tarea
- *Progress*: Parámetros que necesitaremos para actualizar la Interfaz Gráfica.

- *Result*: Dato que devolveremos una vez terminada la tarea.

En el siguiente código se presenta la estructura básica de un *AsyncTask*.

```
public class MiTarea extends AsyncTask<Params, Progress, Result> {  
}
```

Para poder actualizar la interfaz gráfica del *View*, dentro de cada ciclo del *AsyncTask* nos valemos de los siguientes métodos:

- *doInBackground()*: Este método será el encargado de realizar la tarea en segundo plano, por lo que se ejecuta en otro hilo, en este método no podremos modificar la interfaz gráfica de la aplicación. Para modificar la interfaz gráfica usaremos los tres métodos siguientes.

```
protected Result doInBackground(Params... p) {  
}
```

- *onPreExecute()*: Este método se ejecutará antes de *doInBackground*, por lo que podremos modificar la interfaz para indicar el comienzo de la tarea.

```
protected void onPreExecute() {  
}
```

- *onProgressUpdate()*: Este método permitirá actualizar la interfaz mientras se está ejecuta la tarea asíncrona. Para ello, desde el método *doInBackground()* deberemos llamar al método *publishProgress()* y pasarle los parámetros.

```
protected void onProgressUpdate (Progress... values){  
}
```

- *onPostExecute()*: Se ejecutará cuando finalice nuestra tarea asíncrona o tras la finalización del método *doInBackground()*.

```
protected void onPostExecute (Result result){  
}
```

Finalmente para actualizar el *View* dentro del método *onProgressUpdate()* debemos valernos del método *invalidate()*.

3.3.3 Reproducción de sonido [31]

Para la reproducción de sonido, existe la librería *MediaPlayer* dentro del paquete de *android.media*, el cual recibe como parámetro el contexto de ejecución y el identificador del recurso a reproducir, como se puede apreciar en el siguiente ejemplo:

```
MediaPlayer mp = MediaPlayer.create(context, identificador_recurso);  
mp.start();
```

Para la reproducción del sonido se invoca al método *start()* y para detener la reproducción al método *stop()*.

3.3.4 Desarrollo Html5 [23]

El desarrollo integral del mundo eléctrico y el marco central del mundo verde se utilizoHTML5, CSS3 yJavaScript, para posteriormente desplegarla dentro de un *WebView* en Android. Para facilitar el desarrollo web se encontró una herramienta llamada SenchaAnimator, que nos permite realizar animación y poder exportarlos aHTML5, CSS3 yJavaScript, obteniendo una aplicación híbrida, a continuación se procederá a detallar la herramienta SenchaAnimator.

SenchaAnimator es un software para Mac, Windows yGNU/Linux, que permite trabajar en una línea de tiempo y realizar animaciones, sin tener que escribir código directamente, pero en caso de necesitarlo da la facilidad de incluirlo.

Con SenchaAnimator, se puede animar texto e imágenes con transiciones suaves y profesionales, botones de diseño con degradados y un control completo de la línea de tiempo. El producto desarrollado es compatible con todos los estándares web y siguiendo conceptos de desarrollo muy parecidos a los de Flash.

Características:

- El desarrollo de la animación es limpio y legible
- Los usuarios tienen un control completo sobre la salida HTML5.
- Produce proyectos que son compatibles con cualquier red móvil, así como *ORMMA*¹³ que es un estándar de la publicidad en redes móviles.
- Obtener en tiempo real información visual y exportar el producto final sin escribir una sola línea de código.
- Crea proyectos que funcionan en todos los navegadores modernos que soporten CSS3.
- Optimiza el código para los dispositivos móviles como Android 2.3+, Apple iOS 4+, BlackBerry, etc.
- Es perfecto para juegos simples y animaciones respaldadas por JavaScript.

A continuación se procederá a detallar la instalación y la forma de utilizar esta herramienta para la creación de animaciones, sin más preámbulo se detallara la instalación de SenchaAnimator en Windows 7 Ultimate.

¹³ORMMA: Contiene un conjunto común de normas para la publicación de anuncios enriquecidos[24].

1. Nos descargamos la herramienta del sitio web de Sencha <http://www.sencha.com/products/animator/download/>, es necesario registrarse para poder utilizar la herramienta por 30 días, al momento de redactar, la versión de Sencha Animator última es 1.3 y el costo es de \$99 dólares.

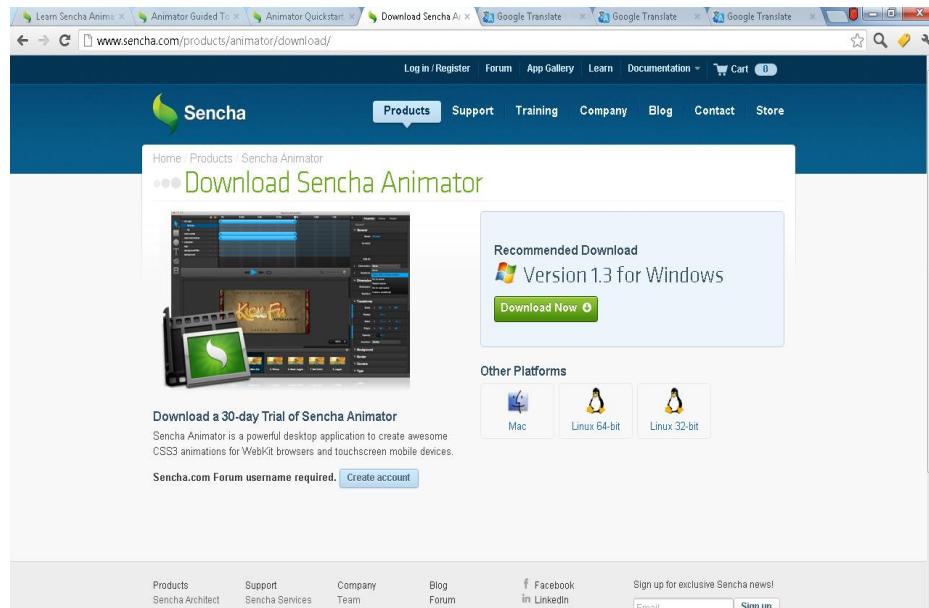


Ilustración 37 Sitio web de descarga de SenchaAnimator

2. Una vez descargado procedemos a la instalación, damos doble clic sobre el archivo descargado, seleccionamos el idioma de instalación y damos clic en siguiente.

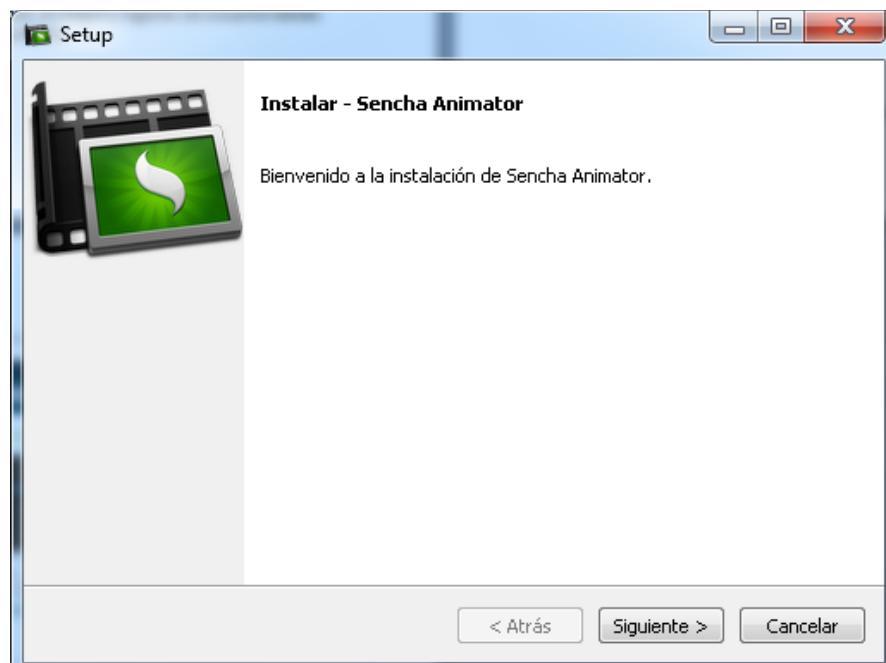


Ilustración 38 Instalación de SenchaAnimator

3. Escogemos el directorio de instalación y damos clic en siguiente.

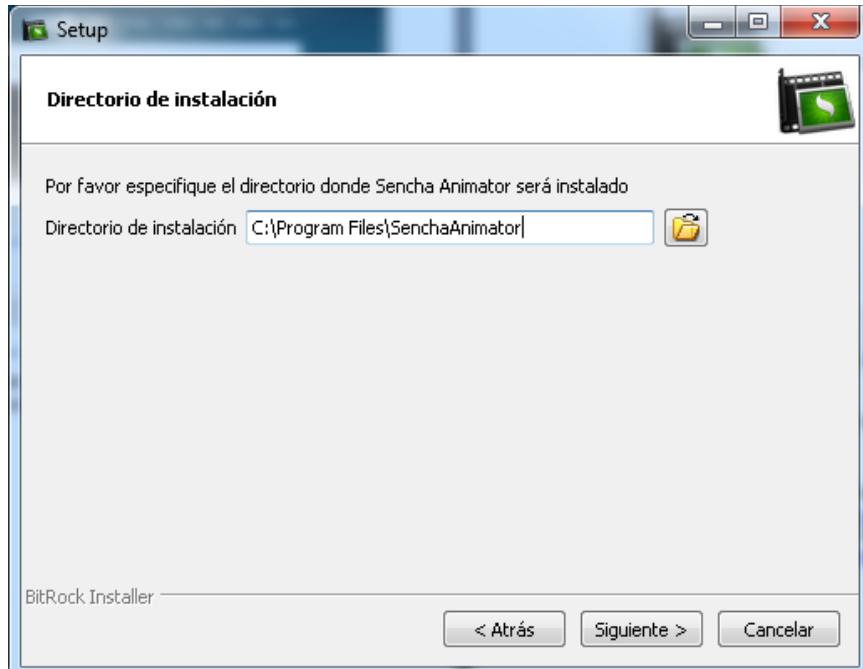


Ilustración 39 Directorio de instalación de Sencha Animator

4. Empezara el proceso de instalación, una vez terminada la instalación damos clic en terminar.



Ilustración 40 Instalación terminada de Sencha Animator

- Procedemos a iniciar Sencha Animator, como se menciono anteriormente para poder utilizar los 30 días de prueba debemos crear un usuario en la página principal de Sencha Animator, para el cual ingresamos un nombre de usuario, contraseña, correo electrónico, y aceptamos las condiciones de licencia.
- Ingresamos el usuario y contraseña creada en el paso anterior para la poder utilizar la aplicación, se nos visualiza el entorno de trabajo de Sencha Animator.

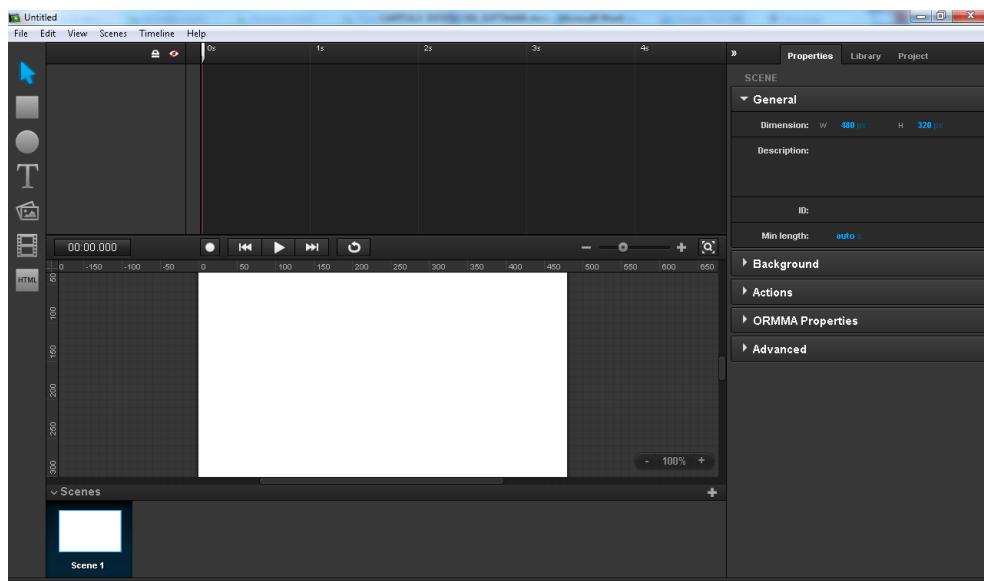


Ilustración 41 Pantalla principal de Sencha Animator

3.3.4.1 Creación de una animación en Sencha Animator

A continuación se dará una introducción de cómo utilizar esta herramienta para crear y exportar la animación HTML5, CSS3 y JavaScript.

- Primero debemos crear un nuevo proyecto, para esto nos dirigimos al Menú *File > New Project*, se nos visualizara la ventana de configuración del nuevo proyecto. Para esto escogemos la dimensión del proyecto y agregamos soporte para Android 2.x, damos clic en *Create*.

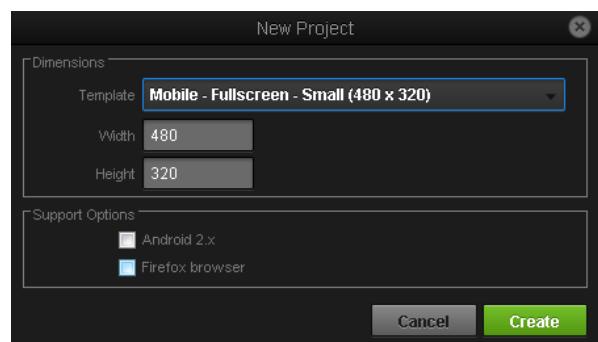


Ilustración 42 Configuración del nuevo Proyecto de Android

2. Guardamos el nuevo proyecto creado para eso nos dirigimos al Menú *File >SaveProject*,
3. Escogemos el directorio en donde se almacenará el proyecto y los archivos adjuntos, le asignamos un nombre al proyecto y damos clic en *Save*, es recomendable guardar el proyecto frecuentemente.
4. Para agregar nuevos elementos al proyecto tenemos en la parte superior izquierda un panel en donde podemos escoger un cuadrado, circunferencia, video, imagen, texto, audio e incrustar código HTML.



Ilustración 43 Panel de objetos de SenchaAnimator

5. En la parte superior derecha tenemos un panel en donde podemos modificar la propiedades de cada uno de los objetos, entre las cuales tenemos dimensiones, transformaciones, fondos, bordes, traslaciones, etc.

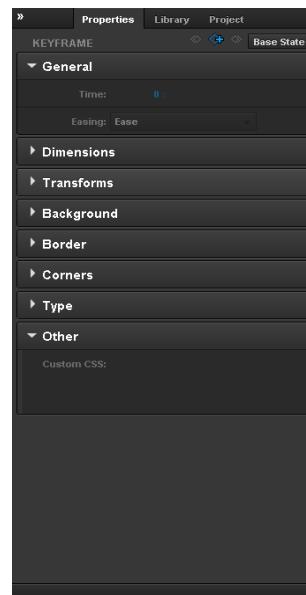


Ilustración 44 Panel de propiedades de Sencha Animator

6. En la parte superior tenemos un panel de tiempo, donde podemos agregar nuevos *frames*, lo que nos permite realizar las transacciones, animaciones o modificaciones de objetos utilizando los *frames* según la línea de tiempo.

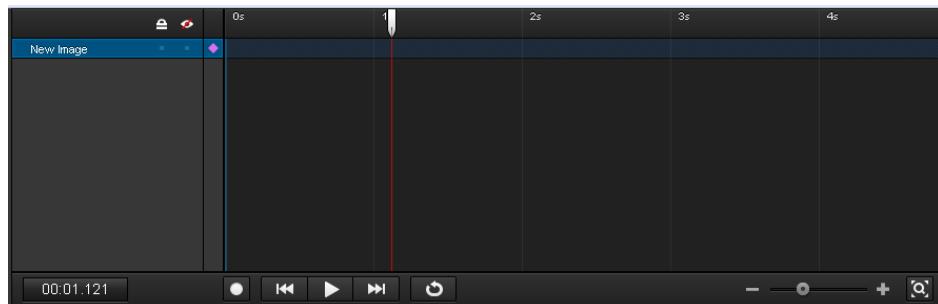


Ilustración 45 Panel línea de tiempo de Sencha Animator

7. En la parte central tenemos nuestro espacio de trabajo. Para agregar un nuevo objeto solo debemos arrastrar del panel de objetos al espacio de trabajo, podemos agregar varios objetos en el mismo espacio y realizar las modificaciones creando o modificando los *frames*, teniendo una visualización previa del proyecto.

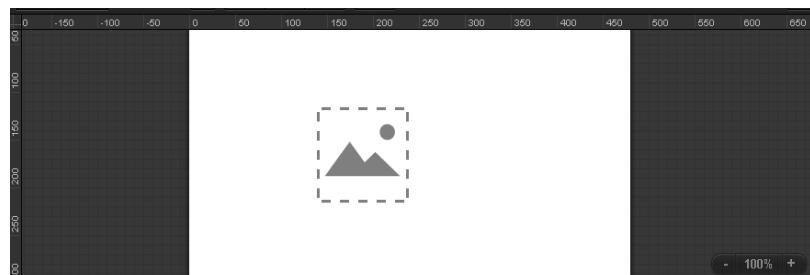


Ilustración 46 Espacio de trabajo de Sencha Animator

8. En la parte inferior tenemos cada una de las escenas creadas en nuestro proyecto. Una escena es un parte del proyecto, por consecuente un proyecto está compuesto de varias escenas.

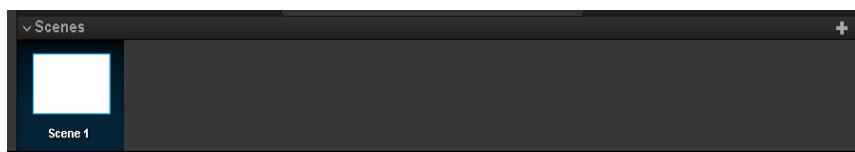


Ilustración 47 Panel de escenas de Sencha Animator

9. Por último, tenemos el panel de control de la línea de tiempo, en el cual nosotros podemos controlar el tiempo y obtener una pre-visualización del proyecto.



Ilustración 48 Panel de control de tiempo de Sencha Animator

CAPÍTULO 4

4 IMPLEMENTACIÓN, DOCUMENTACIÓN Y PRUEBAS

4.1 Implementación y selección de los módulos de aprendizaje.

Se seleccionaron el mundo eléctrico y el mundo verde para el desarrollo de la aplicación para dispositivos móviles, el mundo eléctrico se desarrolló por completo utilizando tecnología web, es decir, solo se empleó para el desarrollo HTML5, CSS3 y JavaScript, el resultado no fue el esperado por el equipo de Pequeñas y Pequeños Científicos. El principal inconveniente es que el desarrollo multimedia de movimientos no soporta cambios de imágenes de forma fluida como se esperaba que tuvieran los personajes de la aplicación, aquí es donde podemos apreciar las transiciones o cambios de imágenes cuando trabajamos en tiempos de milisegundos. Esto se genera porque la animación no se ejecuta directamente en la máquina virtual de Android Dalvik, produciendo un mayor consumo de recursos al tener que primero renderizar HTML5 y CSS3 e interpretar el código JavaScript por cada ejecución, por consiguiente es más costoso que ejecutar una aplicación nativa que ya está previamente compilada dentro de los archivos .dex. En las siguientes ilustraciones capturadas cada 200ms se puede apreciar la transición del personaje en el mundo eléctrico durante un lapso de 600ms, obteniendo como resultado la evidente falta de fluidez en sus movimientos.



Ilustración 49 Entrada de la Doctora Atómica en el Mundo Eléctrico en 200ms.

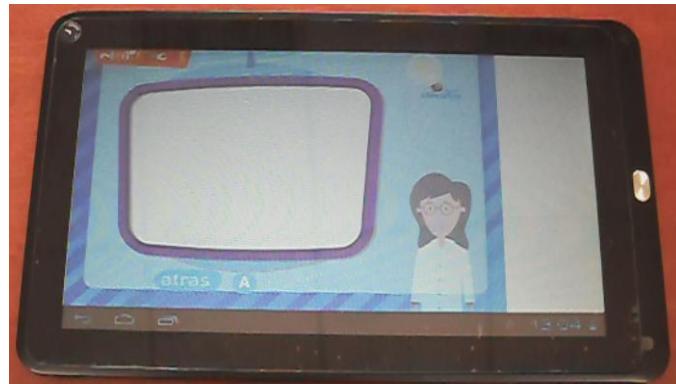


Ilustración 50 Entrada de la Doctora Atómica en el Mundo Eléctrico en 400ms.



Ilustración 51 Entrada de la Doctora Atómica en el Mundo Eléctrico en 600ms.

Sin embargo, los dispositivos móviles tanto en hardware como software se están renovando continuamente, es decir, los nuevos dispositivos móviles ya cuentan con mayores recursos de hardware y nuevas características implementadas en software. Entre estas nuevas características se puede resaltar que las nuevas versiones de Android y IOS ya no integran soporte para flash [33], consecuentemente están optando por HTML5 para la creación de animaciones y multimedia, resultando que esta brecha de compatibilidad entre HTML5 y el consumo de recursos se esté dejando al margen. Consecuentemente, HTML5 sigue en proceso de desarrollo, implementándose nuevas funciones y características. Por consiguiente, no se puede dejar de lado la utilización de HTML5 para la creación de aplicaciones móviles.

Con lo antes expuesto, en el desarrollo del mundo verde se optó por la implementación de aplicaciones híbridas potenciando así las dos tecnologías. En consecuencia, se diferenciaron dos espacios de trabajo, en el primero que corresponde al panel central, se utilizó código HTML5, CSS3 y JavaScript para la visualización del tema y el segundo espacio de trabajo que es la animación de los personajes, fondo, menús y reproducción de sonido se empleó tecnología nativa Android. El resultado fue satisfactoriamente aprobado por el equipo de Pequeñas y Pequeños Científicos. A continuación se pueden apreciar ilustraciones que se tomaron cada 200ms por 600ms obteniendo como resultado:



Ilustración 52 Entrada del Árbol Calíptro en el Mundo Verde en 200ms



Ilustración 53 Entrada del Árbol Calipto en el Mundo Verde en 400ms



Ilustración 54 Entrada del Árbol Calipto en el Mundo Verde en 600ms

Cabe recalcar que el desarrollo del mundo verde fue completamente parametrizado, por lo que en un futuro desarrollo del mundo eléctrico, magnético y digital se pueden reutilizar las clases previamente desarrolladas para minimizar, agilizar y optimizar el tiempo de desarrollo.

4.2 Desarrollo de pruebas con los dispositivos móviles en el laboratorio

Las pruebas realizadas en el laboratorio cumplieron con los requerimientos funcionales y no funcionales antes expuestos, entre estas podemos mencionar que se realizaron pruebas con los siguientes tablets:

Tablet – Coby

- Memoria: 8Gb
- Pantalla: LCD 10.1 pulgadas.
- Procesador: 1Ghz (Corex A8)
- Memoria RAM: 1GB.
- Sistema Operativo: Android 4.0.2 (IceCreamSandwich)



Ilustración 55 Dispositivo Móvil de Prueba Coby.

Tablet –Samsung GalaxyTab 2.

- Memoria: 4 Gb.
- Pantalla: LCD 7.0 pulgadas.
- Procesador: Dual-Core 1 Ghz
- Memoria RAM: 1 GB.
- Sistema Operativo: Android 4.0.2 (Ice Cream Sandwich)



Ilustración 56 Dispositivo Móvil de Prueba Samsung GalaxyTab 2.

Tablet - Asus

- Memoria: 16 GB.
- Pantalla: LCD 10.61 pulgadas.
- Procesador: Dual-Core 2.1 Ghz
- Memoria RAM: 1GB.
- Sistema Operativo: Android 3.2 (HoneyComb).



Ilustración 57 Dispositivo Móvil de Prueba Asus TF101

Cabe mencionar que los *tablets* en donde se realizaron las pruebas de laboratorio fueron proporcionados por la Universidad Politécnica Salesiana.

4.3 Desarrollo de pruebas con niños y niñas.

El desarrollo de pruebas con los niños y niñas fueron realizadas con los estudiantes de la Escuela “Carlos Crespi” y “Francisca Dávila”, obteniendo como resultado una satisfactoria acogida por los niños y niñas, en donde mostraron un mayor interés en la presentación del curso Pequeñas y Pequeños Científicos. Para esto se realizó una encuesta a 18 niños y niñas. Esta encuesta se ha adjuntado en el Anexo 2.

A partir de la encuesta realizada se obtuvo las siguientes gráficas estadísticas.

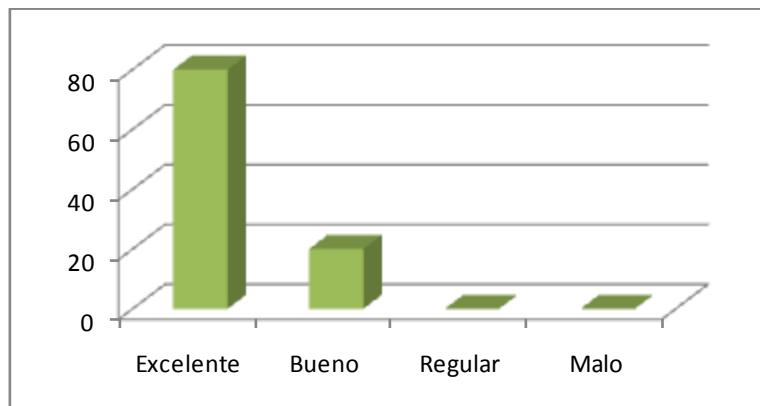


Ilustración 58 Resultado de la pregunta ¿Cómo fue la experiencia de utilización de los *tablets* para el aprendizaje multimedia del mundo verde?

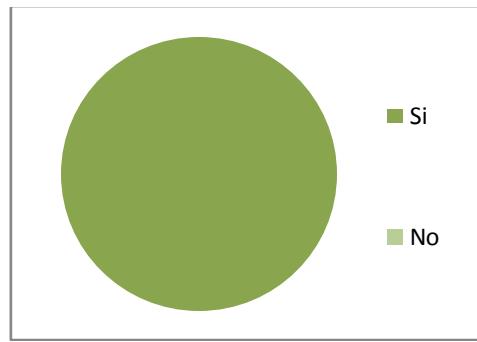


Ilustración 59 Respuesta a la pregunta. ¿Aumentó el grado de interés de aprendizaje utilizando los *tablets*?

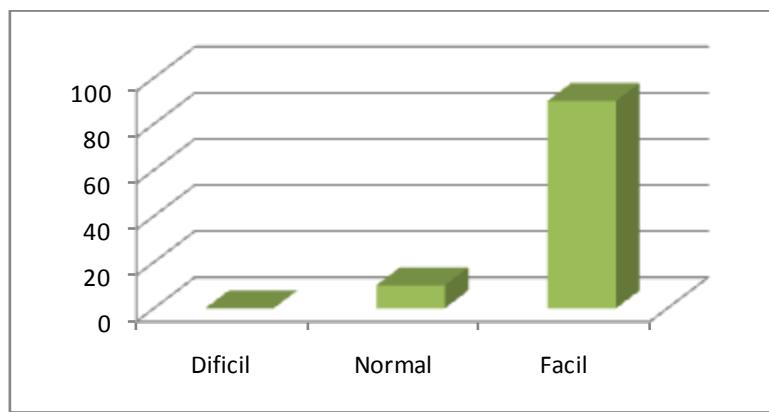


Ilustración 60 Respuesta a la pregunta. ¿La utilización de la aplicación del mundo verde en los *tablets* fue?

Como se puede apreciar en las estadísticas, aumentó notablemente el interés de aprendizaje del mundo verde utilizando los *tablets*, ya que el diseño empleado en la aplicación esta desarrollado para niños y niñas de 8 años en adelante.

Finalmente, presentamos algunas ilustraciones en la que los niños y niñas están utilizando las *tablets* con el aplicativo móvil del mundo verde.



Ilustración 61 Pruebas con Niñas de la Escuela "Francisca Dávila".



Ilustración 62 Pruebas con las Niñas e Instructores.

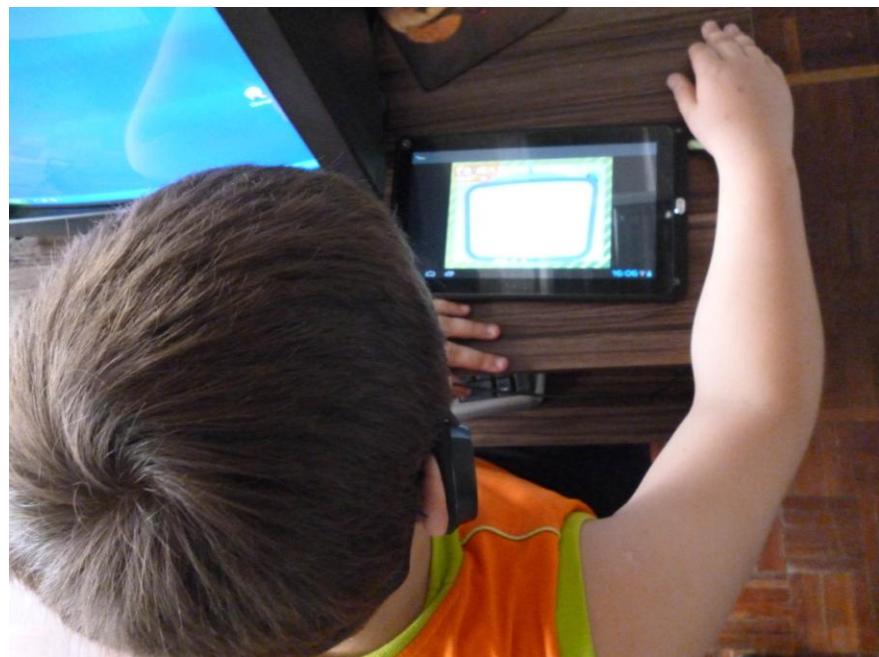


Ilustración 63 Prueba con un estudiante de la Escuela "Carlos Crespi" con la tablet COBY.



Ilustración 64 Pruebas con Niños de la Escuela "Carlos Crespi".

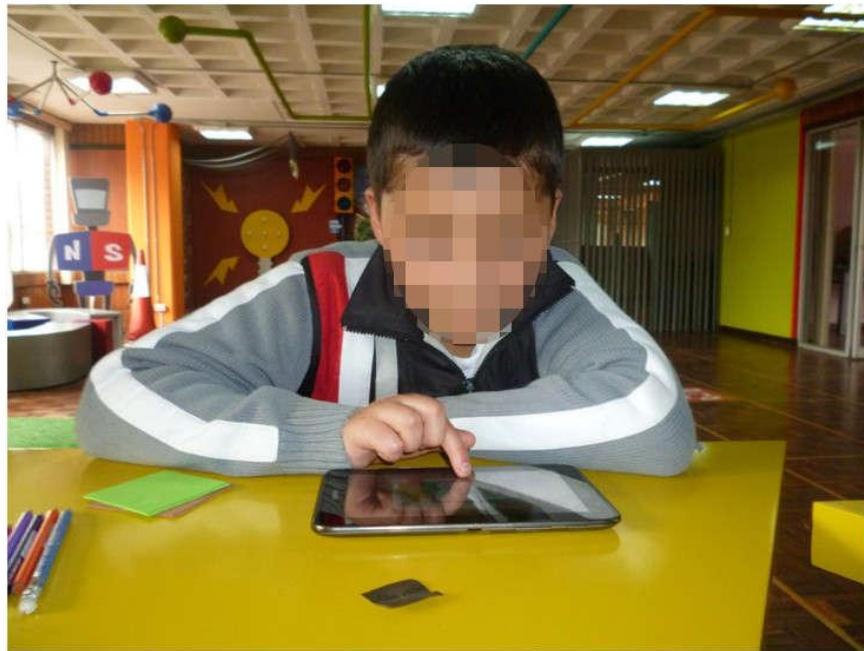


Ilustración 65 Prueba con un estudiante de la Escuela “Carlos Crespi” con la *tablet* Samsung GalaxyTab 2.

4.4 Documentación.

Se procederá a detallar cuáles son los pasos necesarios para que un usuario pueda descargarse la aplicación desde la tienda de *Google Play*, instalarlo en el dispositivo y por último, se indicará el modo de utilizar la aplicación del Mundo Verde, sin más preámbulo se procederá a detallar el manual de Usuario.

4.4.1 Descarga e Instalación de la aplicación.

- Primeramente debemos descargar la aplicación desde la tienda de *Google Play*, para ello nos dirigimos al menú de aplicaciones e ingresamos a nuestro gestor de descargas, en este caso utilizaremos el *Play Store*.

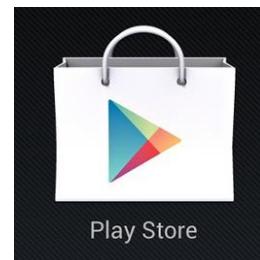


Ilustración 66 Gestor de descargas de aplicaciones *Play Store* [35].

- En el gestor de descargas introducimos el nombre de la aplicación, ingresamos el texto “PequeCiencia”. Se desplegarán todas las aplicaciones que coincidan con la búsqueda, en este caso escogeremos la primera coincidencia. Seleccionamos la aplicación mundo verde como se muestra en la ilustración 67.

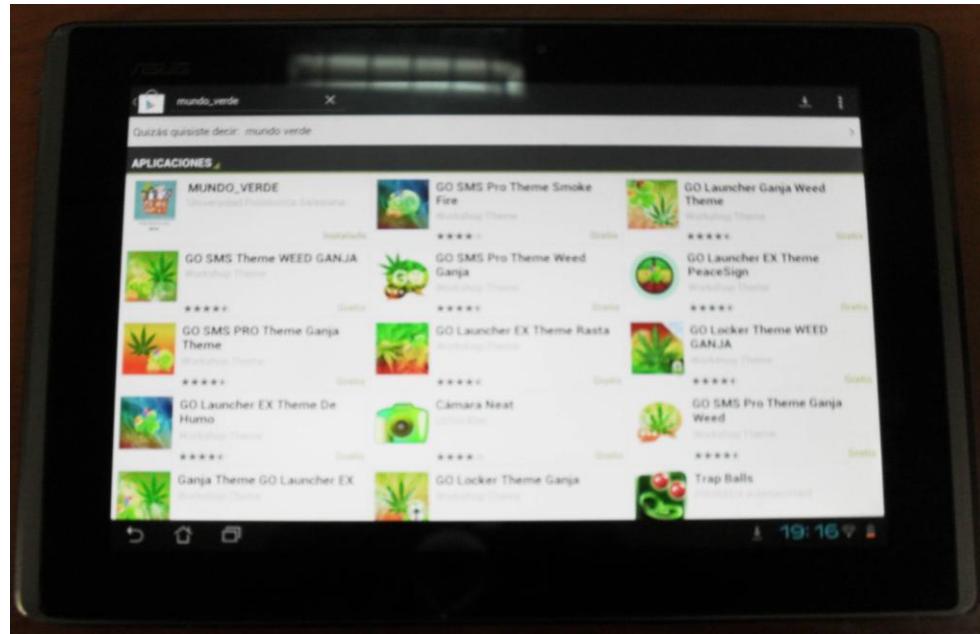


Ilustración 67 Búsqueda de la aplicación en Play Store

- Posteriormente se nos presenta una descripción de la aplicación Mundo Verde, como se muestra en la ilustración 68, presionamos la opción Instalar.

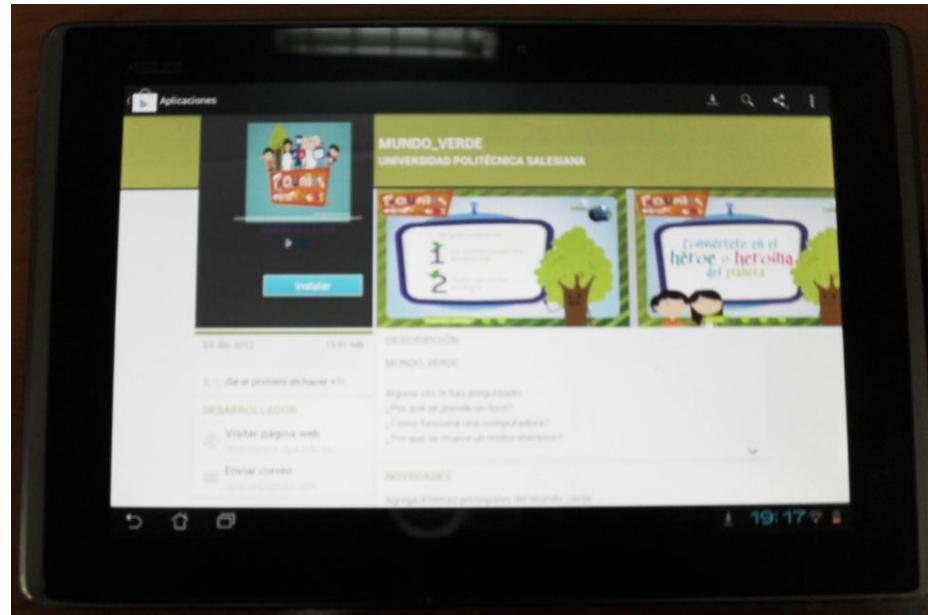


Ilustración 68 Descripción de la Aplicación del Mundo Verde

- Procedemos a Aceptar y Descargar la aplicación, cabe recalcar que no se necesita ningún tipo de permisos adicionales.

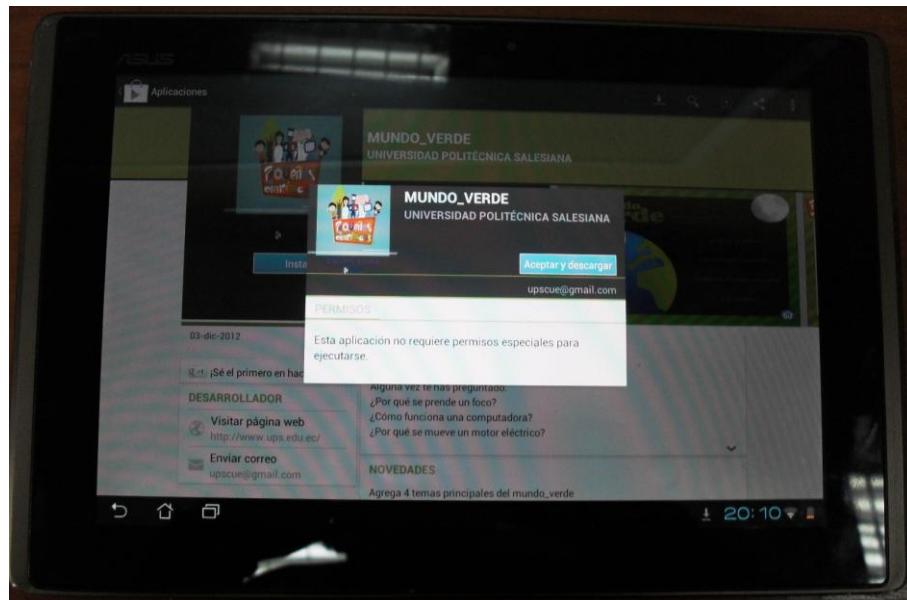


Ilustración 69 Aceptar y Descargar la aplicación del Mundo Verde

- Una vez completada la descarga de la aplicación, se procederá a instalar automáticamente en el dispositivo, finalmente presionamos la opción Abrir.

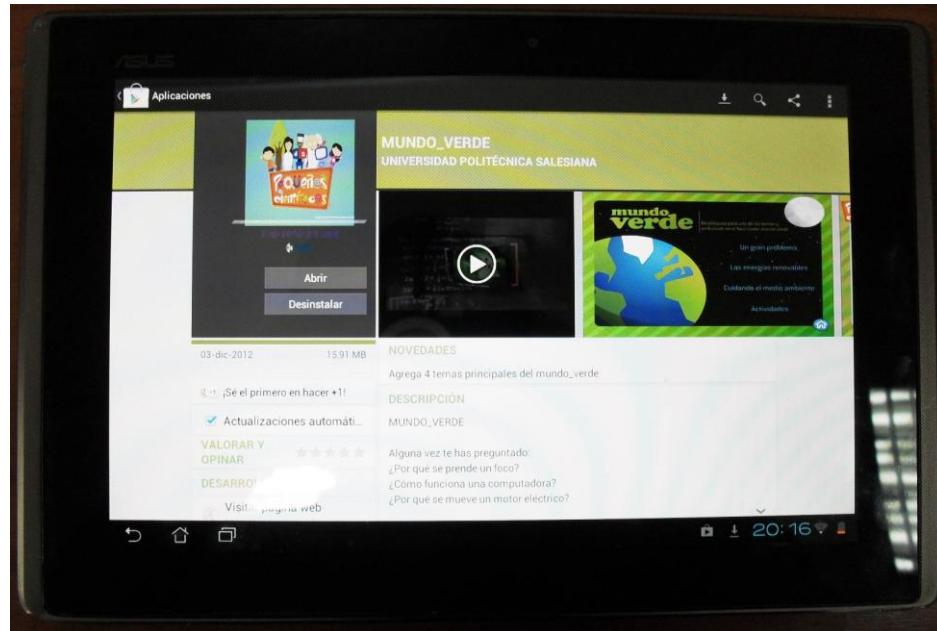


Ilustración 70 Instalación completa de la aplicación Mundo Verde

4.4.2 Manual de Usuario

- Una vez instalada, nos dirigimos al menú de aplicaciones y presionamos en el ícono de la aplicación del Mundo Verde, el ícono se muestra en la ilustración 71.



Ilustración 71 Ícono de la Aplicación del Mundo Verde.

- Se nos presenta el menú principal, el cual está compuesto por cuatro temas como se muestra en la ilustración 72, presionamos el tema de aprendizaje a tratar:



Ilustración 72 Menú Principal del Mundo Verde

- Se nos presenta los subtemas, si deseamos regresar al menú principal presionamos la opción mundo verde, caso contrario presionamos el subtema a estudiar.

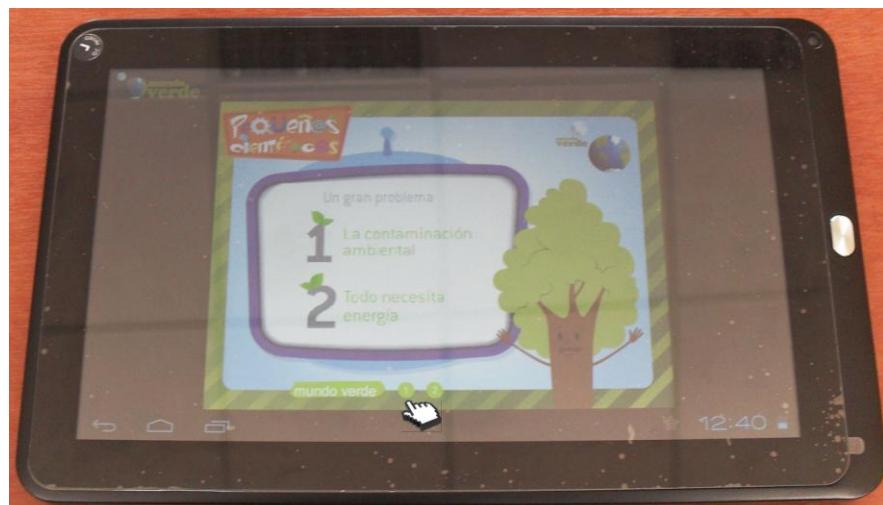


Ilustración 73 Menú de subtemas del Mundo Verde.

- Finalmente, se nos presentan los literales del subtema seleccionado en el paso anterior, para regresar al menú de subtemas presionamos en la opción atrás, caso contrario presionamos ordenadamente cada uno de los literales.



Ilustración 74 Tramos del Mundo Verde

4.5 Publicación de la aplicación en la tienda de Google.

Luego del desarrollo de la aplicación del Mundo Verde, el siguiente paso es la publicación en la tienda de *Google Play*. Todo este proceso se realizará ingresando información de la Universidad Politécnica Salesiana, con el objetivo de que este registro sirva para que futuros proyectos desarrollados dentro de la universidad se publiquen. A continuación se detallan los pasos para la creación de una cuenta desarrollador en Google:

1. Debemos tener creada una cuenta en Gmail, por lo que procederemos a crearla, para eso nos dirigimos a la siguiente dirección web <https://accounts.google.com/SignUp?service=mail&continue=https%3A%2F%2Fmail.google.com%2Fmail%2F<mpl=default> y seguimos estos pasos:
2. Ingresamos los datos personales para la creación de la cuenta, como se puede apreciar en la ilustración 75.

Nombre	Universidad	Politecnica Salesiana	
Nombre de usuario	upscue	@gmail.com	
Contraseña		
Confirma tu contraseña		
Fecha de nacimiento	04	Agosto	1994
Sexo	Otro		
Teléfono móvil	+593-7-2862213		
Tu dirección de correo electrónico actual	dquisi@hotmail.com		

Ilustración 75 Creación de una cuenta en Gmail.

3. Aceptamos las condiciones de servicio y las políticas de privacidad de Google y procedemos a dar clic en siguiente paso.

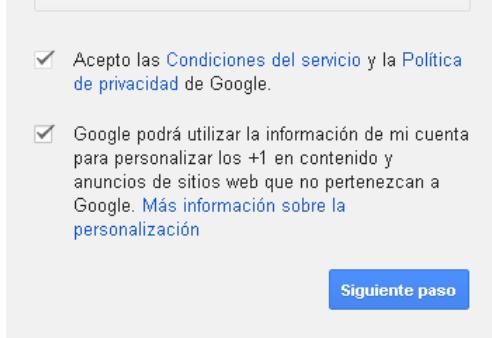


Ilustración 76 Condiciones de Servicios y Políticas de Seguridad de Google.

4. Se nos presenta una pantalla de bienvenida a Gmail, damos clic en Ir a Gmail.

¡Te damos la bienvenida!

Ahora puedes buscar, crear y colaborar en muchos productos de Google.

Tu nueva dirección de correo electrónico es upscue@gmail.com.

Gracias por crear una cuenta. ¡Diviértete!

[Ir a Gmail](#)

Ilustración 77 Pantalla de Bienvenida a Gmail.

5. Finalmente se visualizará la bandeja de entrada de la cuenta creada.

Posteriormente creamos un perfil de desarrollador en *Google Play*, para eso nos dirigimos a la siguiente dirección web <https://play.google.com/apps/publish/signup> y seguimos los siguientes pasos:

1. Creamos un perfil de desarrollador, para ello ingresamos los datos básicos del desarrollador, como se muestra en la ilustración 78, damos clic en Seguir.

The screenshot shows the Google Play Developer Console sign-up process. At the top, there's a logo for 'Google play' and 'ANDROID DEVELOPER CONSOLE'. Below it, a section titled 'Procedimientos iniciales' (Initial procedures) says: 'Para poder publicar software en Google Play, debes seguir estos tres pasos:' (To be able to publish software in Google Play, you must follow these three steps:)

- Crear un perfil de desarrollador
- Acepta el [Acuerdo de distribución para desarrolladores](#).
- Pagar una cuota de registro (25,00 US\$) con tarjeta de crédito (mediante Google Checkout)

Below this, a section titled 'Especificación de detalles' (Specification details) asks: 'Tu perfil de desarrollador determina cómo te ven los clientes en Google Play.' (Your developer profile determines how clients see you in Google Play.) It contains fields for:

- Nombre del desarrollador: Universidad Politécnica S. (Note: 'S.' is in a separate box)
- Dirección de correo electrónico: upscue@gmail.com
- URL del sitio web: http://www.ups.edu.ec/
- Número de teléfono: +593 72862213

There's also a checkbox for 'Actualizaciones por correo electrónico' (Email updates) with the note: 'Incluir el signo más, el código del país y el de área (por ejemplo, +1-650-253-0000). [Más información](#)' (Include the plus sign, country code, and area code (for example, +1-650-253-0000). [More information](#))

Seguir »

Ilustración 78 Creación del perfil desarrollador en *Google Play*.

- Leemos y aceptamos el acuerdo de distribución para desarrolladores de Google, damos clic en Acepto y Continuar.



Ilustración 79 Acuerdo de distribución para desarrolladores

- Por último, debemos pagar la cuota de registro, para ello especificamos nuestro nombre, dirección de facturación y la tarjeta de crédito para el pago de 25 dólares, como se muestra en la ilustración 80.

Ilustración 80 Pago de cuota de registro.

Luego de crear la cuenta desarrollador se procederá a publicar la aplicación en *Google Play*, esta aplicación se podrá descargar gratuitamente desde cualquier parte del mundo, para esto nos dirigimos a la siguiente dirección web <https://play.google.com/apps/publish>. A continuación se detallan los pasos:

1. Se nos presentará el panel de administración de aplicaciones, el cual nos informa de todas las aplicaciones publicadas. Para subir una nueva aplicación damos clic en “Añadir nueva aplicación”.



Ilustración 81 Panel de administración de publicaciones en Android.

2. Se nos presenta una ventana, en donde seleccionamos el idioma de la publicación, nombre de la aplicación y damos clic en subir APK.

Ilustración 82 Ventana para subir el proyecto de Android.

3. Escogemos la aplicación empaquetada que se subirá al servidor, con extensión .apk.



Ilustración 83 Selección del .apk

4. La aplicación se empezaría subir al servidor de Google, el tiempo de espera depende del tamaño de la aplicación.

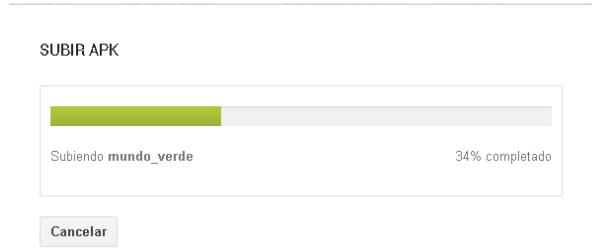


Ilustración 84 Cuadro de progreso de subir la aplicación al servidor.

5. Posteriormente se nos presenta la ventana con la información de la aplicación subida, como la versión, dispositivos admitidos y dispositivos excluidos.

Google play | Developer Console PREVIEW Universidad Politécnica Salesiana upscue@gmail.com Cerrar sesión Envíanos tus comentarios Cambiar al diseño anterior

PEQUECIENCIA UPS - MUNDO VERDE - com.ups.mundo_verde Borrador

APK Subir nuevo APK

ARCHIVO APK ACTUAL subido el 23/11/2012 14:59:27

Código de la versión	Nombre de la versión	Dispositivos admitidos	Dispositivos excluidos
1	1.0	787 Ver dispositivos admitidos	0 Administrar dispositivos excluidos

Aún no hemos incorporado asistencia para **archivos de expansión APK** en el nuevo diseño. Vuelve a usar el diseño anterior para editar archivos de expansión APK.

Ilustración 85 Información de la aplicación subida al servidor.

6. A continuación debemos especificar la información del proyecto, por lo que debemos dar clic en “Entrada en *Play Store*”, en donde podemos especificar con mayor detalle las características de la aplicación, con los siguientes campos:

- INFORMACIÓN DEL PRODUCTO
 - a. Título.
 - b. Descripción.
 - c. Texto Promocional (opcional).
 - d. Cambios Recientes (opcional).
- RECURSOS GRÁFICOS
 - a. Capturas de pantalla.
 - b. Icono de la aplicación.

- c. Gráfico de las Características (opcional).
 - d. Gráfico Promocional (opcional).
 - e. Video Promocional (opcional).
- CATEGORIZACIÓN
 - a. Tipo de la aplicación.
 - b. Categoría.
 - c. Clasificación de contenido.
 - DETALLES DEL CONTACTO.
 - a. Sitio web.
 - b. Correo Electrónico.
 - c. Teléfono (opcional).

En las siguientes ilustraciones se puede apreciar la configuración antes mencionada para el proyecto del Mundo Verde de Pequeñas y Pequeños Científicos.



Ilustración 86 Especificaciones de la Información del Mundo Verde.

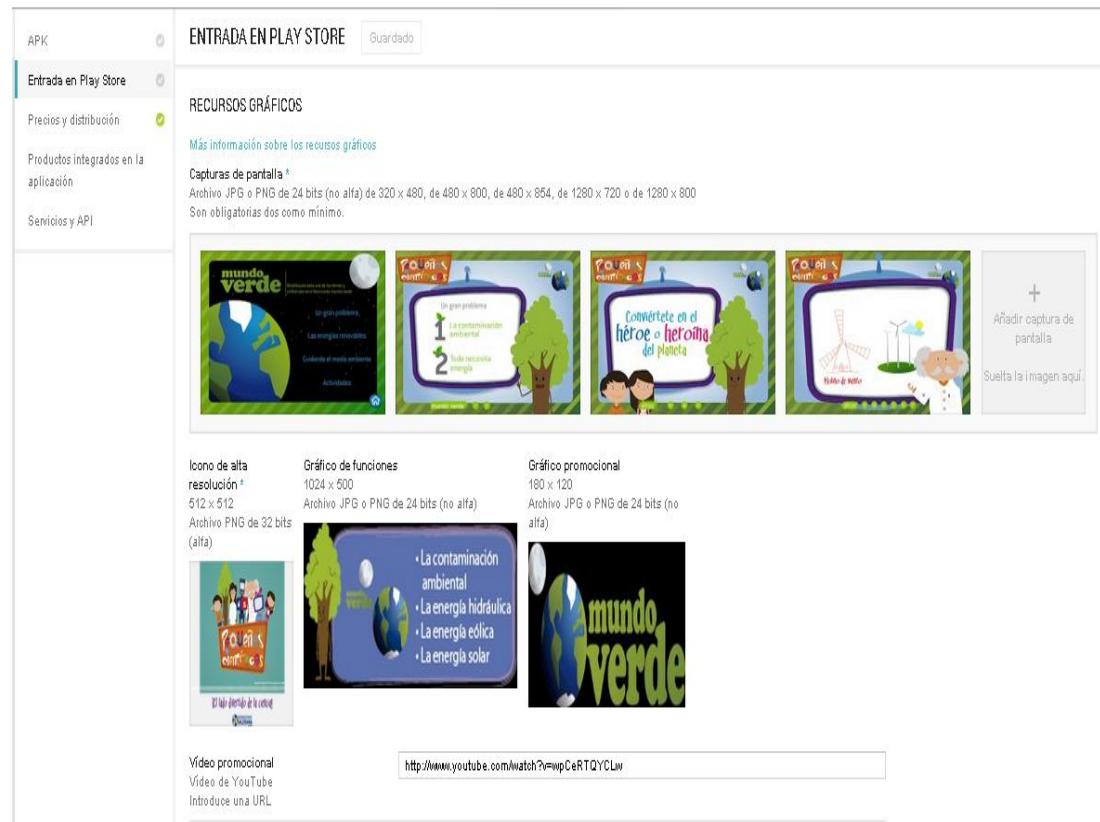


Ilustración 87 Especificaciones Graficas del Mundo Verde.

This screenshot shows the continuation of the Play Store submission process. The left sidebar remains the same. The main area is titled 'CURSO...' (Course...). It includes sections for 'CATEGORIZACIÓN' (Classification), 'DETALLES DEL CONTACTO' (Contact Details), and 'POLÍTICA DE PRIVACIDAD' (Privacy Policy). In the 'CATEGORIZACIÓN' section, 'Aplicaciones' (Applications) is selected under 'Tipo de aplicación' (Type of application) and 'Educación' (Education) under 'Categoría' (Category). The 'DETALLES DEL CONTACTO' section contains fields for 'Sitio web' (Website) with the value 'http://www.upscue.edu.ec/' and 'Correo electrónico' (Email) with the value 'upscue@gmail.com'. The 'POLÍTICA DE PRIVACIDAD' section has a note about providing a privacy policy URL and a checkbox for indicating no privacy policy is provided.

Ilustración 88 Categorización y detalle de contacto del Mundo Verde.

Para mayor detalle de como especificar el producto dirigirse al sitio web de Soporte de Google (<https://support.google.com/googleplay/android-developer/support/bin/answer.py?hl=es&answer=1078870>), posteriormente damos clic en Guardar.

7. Finalmente procedemos a especificar el precio de la aplicación y en qué países se distribuirá. Esta aplicación será distribuida en todo el mundo de forma gratuita por lo que no necesitamos ninguna configuración adicional y damos clic en Guardar.

The screenshot shows the 'Precios y Distribución' (Prices and Distribution) section of the Google Play Developer Console. At the top, there are two buttons: 'De pago' (Paid) and 'Gratis' (Free). Below them, a note says: 'Para publicar aplicaciones de pago, debes configurar una cuenta de comerciante. Crea una cuenta de comerciante ahora o Más información'. A large list of countries is displayed under the heading 'DISTRIBUIR EN ESTOS PAÍSES' (Distribute in these countries). The list includes: Albania, Alemania, Angola, Antigua y Barbuda, Antillas Neerlandesas, Arabia Saudí, Argelia, Argentina, Armenia, Aruba, and others. Each country has a checkbox next to its name. A 'Mostrar opciones' (Show options) link is visible at the top right of the list. At the bottom of the list, there is a 'CONSEGUIR DISTRIBUCIÓN' (Get distribution) button.

Ilustración 89 Especificación del precio y distribución del Mundo Verde.

8. Al concluir visualizaremos la aplicación configurada y publicada en la tienda de *GooglePlay*.



Ilustración 90 Presentación de la aplicación en *Google Play* del Mundo Verde.

Una característica adicional que nos brinda *Google Play* es que podemos visualizar cuadros estadísticos de descargas de la aplicación.

5 CONCLUSIONES

Se ha obtenido un gran conocimiento sobre desarrollo en dispositivos móviles basados en Android, así como su arquitectura, características, componentes y funcionamiento. Cabe resaltar que el desarrollo en Android es muy parecido al desarrollo en lenguaje Java, ya que incluye algunos *APIs* de este lenguaje, por ello el aprendizaje y el desarrollo se realizó de forma óptima.

Dentro del desarrollo del presente proyecto se investigaron varios procesos de conversión de archivos de formato en flash (SWF) a imágenes animadas (GIF), el principal problema es que el archivo generado era 7 veces mayor que el original. De igual forma, se probó realizar conversiones a videos y el resultado fue similar, teniendo archivos 5 veces mayores que el original. Por último, se realizaron conversiones a formato HTML5, siendo la que mejor resultados dio, para ello se utilizó herramientas de conversión como *Swiffy* de Google que aun esta en versión de prueba y retorna buenos resultados en tamaño de espacio en disco, pero con información encriptada, por lo que no se puede modificar. En consecuencia, ninguna de las técnicas antes expuestas cumplía con los requerimientos mínimos para la ejecución de la aplicación en los dispositivos móviles, ya que los mismos tienen menor capacidad en hardware.

Una de las principales razones por la que se desarrolló la aplicación en Android, es su alto nivel de utilización en plataformas celulares y *tablets*, en consecuencia se permite que se pueda distribuir, modificar y estudiar sin limitaciones, ya que es un sistema completamente libre [14] y a la vez el desarrollador puede decidir sobre su aplicación, si desea publicarla comercialmente o gratuitamente y los lugares en donde se distribuirá.

Se realizó una completa investigación de desarrollo para dispositivos móviles, encontrándose que se pueden implementar aplicaciones utilizando diferentes tecnologías web como HTML5, CSS3 y JavaScript y con ayuda de *frameworks* como JQuery Mobile, Sencha, PhoneGap, etc. Estas tecnologías se tomaron como primer punto de partida, con el objetivo de facilitar el desarrollo de aplicaciones y hacerlas multiplataforma, por ello se decidió desarrollar el módulo del mundo eléctrico en esta tecnología, pero durante su desarrollo se produjeron grandes limitaciones, específicamente al momento de representar la acción de movimiento de los personajes, lo cual no cumplió con las expectativas del equipo de Pequeñas y Pequeños Científicos. En virtud de ello, esta tecnología aun sigue en un proceso de desarrollo, por lo que los dispositivos actuales no contienen las características necesarias para una perfecta ejecución de las mismas ni en hardware ni en software, pero esta brecha se está reduciendo por motivos de avance y actualizaciones tecnológicas.

Por lo antes expuesto, se incursionó en otra línea de desarrollo, optando por un desarrollo híbrido para el módulo del “Mundo Verde”, que combinaba código Android con HTML5. Durante este proceso se pudo observar que la

libreríaOpen-GL solo puede trabajar con imágenes con un tamaño máximo de 2048 x 2048 pixeles, por lo que en algunos casos era necesario redimensionar las mismas. Dentro de este proceso de desarrollo se implementaron clases parametrizables, para que en un futuro se pueda agregar nuevas funcionalidades o implementar los mundos eléctrico, magnético y digital.

En el desarrollo se presentaron varios problemas entre los que podemos mencionar el manejo de tiempos, por lo que se emplearon hilos asíncronos (*AsyncTask*), el problema al utilizar los *AsyncTask* es que se puede utilizar un máximo de cinco *AsyncTask* ejecutadas en paralelo por aplicación y con un Sistema Operativo Android 3.0 o superior. Por ello, se modelaron hilos asíncronos para manejar varios procesos, como por ejemplo, se utilizó un solo hilo asíncrono para el manejo de los personajes del árbol Calipto y el científico Dr. Voltio.

Con el Mundo Verde se realizaron pruebas en *tablets* de 10.6 ,10 y 7 pulgadas con versiones de Android 4.0.2 y 3.0.2 respectivamente, los cuales fueron facilitados por la Universidad Politécnica Salesiana, en donde se validaron los diferentes requerimientos funcionales y no funcionales conjuntamente con el equipo de Pequeñas y Pequeños Científicos de la Universidad Politécnica Salesiana. En cuanto al proceso de pruebas, estas se realizaron con niños y niñas de las Escuelas “Carlos Crespi” y “Francisca Dávila”, en donde se observó el gran entusiasmo y empeño en el aprendizaje de la ciencia y la tecnología del proyecto desarrollado.

Por último se detallaron los pasos para crear una cuenta de desarrollador en *Google Play*, con el objetivo de publicar la aplicación del mundo verde y futuras aplicaciones que sean desarrolladas dentro de la Universidad Politécnica Salesiana y así promover el desarrollo de aplicaciones móviles.

6 RECOMENDACIONES

Seguir con el proceso de investigación de desarrollo para dispositivos móviles utilizando tecnologías web, ya que empresas como Google y Apple están apostando por HTML5, CSS3 y JavaScript para la realización multimedia o de animaciones, entre estos procesos se puede destacar el uso de *frameworks* como JQuery Mobile, Sencha, Titanium, Phonegap, etc., que facilitan notablemente el desarrollo de aplicaciones móviles, agregando una ventaja más que es la de contar con aplicaciones multiplataforma.

Seguir desarrollando aplicaciones móviles dentro del CIIDI (Centro de Investigación, Desarrollo e Innovación en Ingenierías de la UPS) para diferentes campos aplicativos como la educación, medicina, etc. y publicarlos dentro de la cuenta de la Universidad Politécnica Salesiana, fomentando la innovación y desarrollo de aplicaciones móviles.

7 TRABAJOS FUTUROS

Seguir con el proceso de desarrollo con los mundos eléctrico, magnético y digital, reutilizando código ya desarrollado dentro de la tesis, ya que como se mencionó anteriormente, se desarrolló la aplicación de forma parametrizada, para que se pueda reutilizar el código y así facilitar el desarrollo e implementación en plataformas como IOS y Windows 8.

8 BIBLIOGRAFÍA

- [1] Jimmy Rosario, La Tecnología de la Información y la Comunicación (TIC). Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual, fecha de recuperación 20 de agosto de 2012, <http://www.cibersociedad.net/archivo/articulo.php?art=218>
- [2] TouchTerms.com, fecha de recuperación 20 de agosto de 2012, Touchscreen: <http://www.techterms.com/definition/touchscreen>
- [3] Ing. Elena Durán, Ing. Diego Granja, Informe de Resultados de Proyecto de Vinculación con la Sociedad “Pequeñas y Pequeños Científicos” presentado dentro de la Universidad Politécnica Salesiana, fecha de recuperación 06 de septiembre de 2012
- [4] José Ramón Gómez, Las Tic en la Educación, fecha de recuperación 17 de noviembre de 2012, <http://boj.pntic.mec.es/jgomez46/ticedu.htm>
- [5] LucianArmasu, IDC AndroidJump 68.1% global market share, iOS slips to 16.9%, fecha de recuperación 17 de noviembre de 2012, <http://www.androidauthority.com/idc-android-jumps-to-68-1-global-market-share-ios-slips-16-9-106446/>
- [6] Miguel Angel Alvarez, HTML5 novedades más significativas, fecha de recuperación 07 de agosto de 2012, <http://www.desarrolloweb.com/articulos/que-es-html5.html>
- [7] Rodrigo Mejía Armijo y Diego Escares Venegas, Desarrollo HTML5 para iPad y tabletas, fecha de recuperación 07 de agosto del 2012, <http://habitatweb.mx/desarrollo-html5-para-ipad-y-tabletas>
- [8] Héctor Hiram Guedea Noriega, Introducción a HTML5, fecha de recuperación 07 de agosto de 2012 <http://hectorguedea.com/introduccion-a-html5>
- [9] Gran Angular Blog, Conoce (y prueba) las características de HTML5, fecha de recuperación: 07 de agosto 2012, <http://www.gran-angular.net/conoce-prueba-caracteristicas-html5-presentacion-practica-tecnicas-css3/2010/04/30/>
- [10] Sergio Luján Mora, Talleres sobre HTML5, WCAG 2.0 y web móvil del W3C, fecha de recuperación 07 de agosto de 2012 <http://accesibilidadenlaweb.blogspot.com/2012/05/talleres-sobre-html5-wcag-20-y-web.html>
- [11] W3Schools, HTML5 Introduction, fecha de recuperación 07 de agosto de 2012 http://www.w3schools.com/html5/html5_intro.asp
- [12] jQueryFoundation, BuildingPhoneGapapps withjQuery Mobile, fecha de recuperación 07 de agosto de 2012 <http://jquerymobile.com/test/docs/pages/phonegap.html>
- [13] Adobe Systems Inc. PhoneGap, fecha de recuperación 07 de agosto de 2012 <http://phonegap.com/>

- [14] Jaime Aranaz Tudela, UNIVERSIDAD CARLOS III DE MADRID ESCUELA POLITÉCNICA SUPERIOR, DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES SOBRE LA PLATAFORMA ANDROID DE GOOGLE, Enero de 2009, fecha de recuperación 23 de agosto de 2012, http://e-archivo.uc3m.es:8080/bitstream/10016/6506/1/PFC_Jaime_Aranaz_Tudela_2010116132629.pdf
- [15] Alberto Biches, Desarrollo de aplicaciones móviles multiplataforma, fecha de recuperación 23 de agosto del 2012 <http://albertovilches.com/aplicaciones-moviles-web-vs-nativas-vs-multiplataforma>
- [16] Dosisgeek, Blog Geek, Razones para usar Android, <http://www.tudosisgeek.com/10-razones-para-usar-android-ventajas/>
- [17] HamzaWaqas, Phonegap vs. Native—Comparison basedo verdepth, fecha de recuperación 23 de agosto del 2012, <http://informationastronaut.blogspot.com/2012/05/phonegap-vs-native-comparison-based.html>
- [18] Guillermo Latorre, Los 4 mejores editores de animaciones en HTML5 y CSS3, fecha de recuperación 17 de noviembre de 2012, <http://hachemuda.com/2012/02/los-4-mejores-editores-de-animationes-en-html5-y-css3/>
- [19] ALEGSA, DICCIONARIO DE INFORMÁTICA, fecha de recuperación 17 de noviembre de 2012, <http://www.alegsa.com.ar/Dic/middleware.php>
- [20] Luis Rey Lara, fecha de recuperación 17 de noviembre de 2012, <http://luisreylara.com/2011/11/05/activity-en-android/>
- [21] AndroidDevelopers, PlatformVersions, fecha de recuperación 17 de noviembre del 2012 <http://developer.android.com/about/dashboards/index.html>
- [22] CeluActivo, Introducción a la programación en Android, fecha de recuperación 17 de noviembre de 2012, <http://www.celuactivo.com/2012/08/27/introduccion-a-la-programacion-en-android/>
- [23] Sencha, Sencha Animator, fecha de recuperación 06 de septiembre de 2012, <http://www.sencha.com/products/animator>
- [24] Orrma.org, Open Rich Media Mobile Advertising, fecha de recuperación 06 de septiembre de 2012, <http://code.google.com/p/ormma/>
- [25] Nekotek, Stakeholders, fecha de recuperación 06 de septiembre de 2012, <http://www.nekotek.es/blog/2010/12/stakeholders/>.
- [26] Ing. Natalia Valeria Andriano, Comparación del Proceso de Elicitación de Requerimientos en el desarrollo de Software a Medida y Empaquetado. Propuesta de métricas para la elicitation Universidad Blas Pascal – Universidad Nacional de la Plata, fecha de recuperación 06 de septiembre de 2012 http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Ingenieria_de_Software/Tesis/Andriano_Natalia.pdf,

- [27]Miguel Angel, Androides y Pingüinos, fecha de recuperación 06 de septiembre de 2012, <http://miguelangellv.wordpress.com/2011/01/31/creando-tareas-asincronas-en-android-con-asynctask/>
- [28] Blog OpenAlfa, Cómo dibujar gráficos en Android, fecha de recuperación 06 de septiembre, <http://blog.openalfa.com/como-dibujar-graficos-en-android/>
- [29] Drawable – Android. Programación Aplicaciones para móviles. Universidad Politécnica de Valencia,Drawable fecha de recuperación 06 de septiembre de 2012, <http://www.androidcurso.com/index.php/recursos-didacticos/tutoriales-android/35-unidad-4-graficos-en-android/136-drawable>
- [30] AndroidDevelopers, BitmapFactory, fecha de recuperación 17 de noviembre de 2012, <http://developer.android.com/reference/android/graphics/BitmapFactory.html>
- [31]AndroidDevelopers, Media Playback, fecha de recuperación 17 de noviembre de 2012, <http://developer.android.com/guide/topics/media/mediaplayer.html>
- [32] ALEGSA, DICCIONARIO DE INFORMÁTICA, fecha de recuperación 17 de noviembre de 2012, <http://www.alegsa.com.ar/Dic/motor%20de%20renderizado%20web.php>
- [33]Alkar, XatakaAndroid, fecha de recuperación 17 de noviembre de 2012, <http://www.xatakandroid.com/aplicaciones-android/adios-flash-para-android-gracias-por-intentarlo>.
- [34] Abaestudio, HTML5, el camino que sigue el desarrollo web, fecha de recuperación 17 de noviembre de 2012,<http://www.abaeestudio.com/html5-el-camino-que-sigue-el-desarrollo-web>
- [35] ABOUT XDA DEVELOPERS, Google Play Store fecha de recuperación 03 de noviembre de 2012,<http://www.xda-developers.com/android/google-play-store-pins-not-quite-secure/>
- [36] Gobierno de España, Formación en Red, fecha de recuperación: 07 de agosto de 2012, <http://formacionprofesorado.educacion.es/index.php/es/materiales/45-html5-en-la-educacion?start=2>

ANEXOS

9 ANEXOS

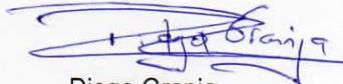
9.1 ANEXO 1: CERTIFICADO DE VALIDACIÓN DE REQUERIMIENTOS.

Cuenca, 30 de noviembre de 2012

Nosotros, ELENA CRISTINA DURÁN LÓPEZ, y DIEGO JOSUÉ GRANJA CARVAJAL, coordinadores del proyecto de vinculación con la sociedad "Pequeñ@s Científic@s" de la Universidad Politécnica Salesiana, certificamos que el módulo "Mundo Verde", producto de la tesis "DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN PARA DISPOSITIVOS ANDROID EN EL MARCO DEL PROYECTO PEQUEÑAS Y PEQUEÑOS CIENTÍFICOS DE LA UNIVERSIDAD POLÍTÉCNICA SALESIANA", desarrollada por Diego Fernando Quisi Peralta, portador de la cédula de identidad 010461646-1, previo a la obtención del título INGENIERO EN SISTEMAS y dirigida por el Ing. Vladimir Robles, ha sido revisado y manifestamos nuestra conformidad como prototipo para la implementación de los otros módulos didácticos que conforman el proyecto Pequeñ@s Científic@s.

Elena C. Durán L.

Elena Durán
Coordinadora del proyecto
"Aula de Ciencia para niños y niñas"



Diego Granja
Coordinador del proyecto
"Aula de Ciencia para niños y niñas"

INVESTIGACIÓN

Turuhuayco 3-69 y Calle Vieja • Casila 2074 • PBX: (593 7) 2862213 Ext. 1254 • Fax: 4088-958
E-mail: investigacion@ups.edu.ec • www.ups.edu.ec • Cuenca - Ecuador

9.2 ANEXO 2: ENCUESTA PRACTICADA A LOS NIÑOS Y NIÑAS

DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN PARA DISPOSITIVOS ANDROID EN EL MARCO DEL PROYECTO PEQUEÑAS Y PEQUEÑOS CIENTÍFICOS DE LA UNIVERSIDAD POLITÉCNICA SALESIANA

ENCUESTA DE UTILIZACION DE TABLETS PARA EL APRENDIZAJE DEL MUNDO VERDE

- 1. ¿Cómo fue la experiencia de utilización de los tablets para el aprendizaje multimedida del mundo verde?**
 - a. Excelente
 - b. Bueno
 - c. Regular
 - d. Malo
- 2. ¿Aumentó el grado de interés de aprendizaje utilizando los tablets?**
 - a. Si
 - b. No
- 3. ¿La utilización de la aplicación del mundo verde en los tabletsfue?**
 - a. Difícil
 - b. Normal
 - c. Fácil

